REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 23.Jan.04	3. REPORT TYPE AND DATES COVERED THESIS

4. TITLE AND SUBTITLE "INVENTORY PLANNING FOR REMANUFACTURING"	5. FUNDING NUMBERS
6. AUTHOR(S) MAJ GAUDETTE KEVIN J	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) INDIANA UNIVERSITY BLOOMINGTON	8. PERFORMING ORGANIZATION REPORT NUMBER CI04-20
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) THE DEPARTMENT OF THE AIR FORCE AFIT/CIA, BLDG 125 2950 P STREET WPAFB OH 45433	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT Unlimited distribution In Accordance With AFI 35-205/AFIT Sup 1	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

20040213 053

14. SUBJECT TERMS	15. NUMBER OF PAGES 202
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

21 January 2004

MEMORANDUM FOR AFIT/CIGK (Maj Clepper)

FROM: AFIT/PA Security and Policy Review

SUBJECT: Your Requests for Public Release Approval,
"Inventory Planning for Remanufacturing"
By Maj Kevin Gaudette

CASE Number: AFIT/PA Case Number TH040106

1. The thesis above has been reviewed and cleared by AFIT/PA for public release with the following stipulation: please add "disclaimer" clause and remove names of family in "Acknowledgement" and date of birth in "Vita" sections.

2. If you have any questions, please call me at DSN: 785-6565 ext 4379.

Pat Wagner

PAT WAGNER
Public Affairs Specialist
Air Force Institute of Technology

SECURITY AND POLICY REVIEW WORKSHEET- Request for Public Release Clearance
(In Accordance With AFI 35-101, Chapter 15)
(Do not use this worksheet to request clearance of software or web pages.)

SUBMITTING ORGANIZATION *(Office Symbol)* TH040106

A. DOCUMENT TYPE *(Documents must be complete including all figures, charts, photographs and text.)*

[ ] ABSTRACT	[ ] JOURNAL ARTICLE	[ ] THESIS	[X] DISSERTATION
[ ] BROCHURES	[ ] NEWS RELEASE	[ ] SPEECH	[ ] OTHER
[ ] CD-ROM	[ ] PHOTO WITH CAPTIONS	[ ] SUCCESS STORY	
[ ] DISPLAY/EXHIBIT	[ ] POSTER SESSION	[ ] TECHNICAL PAPER	
[ ] FACT SHEET	[ ] PRESENTATION WITH TEXT	[ ] TECHNICAL REPORT	

B. TITLE OF DOCUMENT	NO. PAGES
INVENTORY PLANNING FOR REMANUFACTURING	202
C. AUTHOR(S) NAME AND DUTY TITLE	**OFFICE SYMBOL**
KEVIN J. GAUDETTE, MAJ, USAF	AFLMA/LGS

D. FORUM *(Public release clearance is not required for material presented in a closed meeting and which will not be made available to the general public on the Internet or in any print or electronic media)*

DOCUMENT WILL BE	NAME OF CONFERENCE	LOCATION	DATE
[ ] PRESENTED ORALLY			
DOCUMENT WILL BE	**NAME OF PUBLICATION**		**SUBMITTAL DEADLINE**
[X] DISTRIBUTED IN PRINT	DISSERTATION		12 DEC 03

HAVE RELATED DOCUMENTS BEEN PREVIOUSLY CLEARED FOR PUBLIC RELEASE

[X] NO [ ] YES AFIT CASE NUMBER ________

NOTE: If document will be released to a medium outside the Department of Defense, the following disclaimer must be added: *The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.*

FOR AFIT/PA USE ONLY

Dissertation is approved with the following stipulations: please add "disclaimer" clause and remove names of family in "Acknowledgement" and date of birth in "Vita" sections.

SUBMIT TO: Air Force Institute of Technology
Public Affairs (AFIT/PA)
Building 642, Room 2010
2950 P Street
Wright-Patterson AFB OH 45433-7765

FOR MORE INFORMATION

Phone: (937) 255-9354 DSN: 785-9354
FAX: (937) 255-2135 DSN: 785-2135
E-MAIL: afit.pa@afit.edu
WEB: http://www.afit.mil/newsrm01.htm

20 JAN REC'D

20 JAN REC'D

E. NATIONAL SECURITY AND TECHNOLOGY ISSUES

CHECK YES OR NO *(Provide a brief explanation for all positive responses in the SYNOPSIS SECTION below.)*

☐ YES ☒ NO DOES THE MATERIAL HAVE THE POTENTIAL TO BECOME AN ITEM OF NATIONAL OR INTERNATIONAL INTEREST.

☐ YES ☒ NO DOES THE MATERIAL AFFECT NATIONAL SECURITY POLICY OR FOREIGN RELATIONS.

☐ YES ☒ NO DOES THE MATERIAL CONCERN SUBJECTS OF POTENTIAL CONTROVERSY AMONG DOD COMPONENTS OR WITH OTHER FEDERAL AGENCIES.

☐ YES ☒ NO DOES THE MATERIAL CONTAIN TECHNICAL DATA DEVELOPED UNDER CONTRACT OR INDEPENDENTLY DEVELOPED AND CONTROLLED BY THE INTERNATIONAL TRAFFIC IN ARMS REGULATIONS (ITAR) THAT MAY BE MILITARILY CRITICAL AND SUBJECT TO LIMITED DISTRIBUTION, BUT ON WHICH A DISTRIBUTION DETERMINATION HAS NOT BEEN MADE.

☐ YES ☒ NO DOES THE MATERIAL CONTAIN INFORMATION ON NEW WEAPONS OR WEAPON SYSTEMS, SIGNIFICANT MODIFICATIONS OR IMPROVEMENTS TO EXISTING WEAPONS OR WEAPON SYSTEMS, EQUIPMENT OR TECHNIQUES.

☐ YES ☒ NO DOES THE MATERIAL CONTAIN INFORMATION ON NATIONAL COMMAND AUTHORITIES; COMMAND, CONTROL, COMMUNICATIONS, COMPUTERS AND INTELLIGENCE; INFORMATION WARFARE; OR COMPUTER SECURITY.

☐ YES ☒ NO DOES THE MATERIAL CONTAIN INFORMATION ON MILITARY ACTIVITIES OR APPLICATIONS IN SPACE, NUCLEAR WEAPONS, INCLUDING WEAPON-EFFECTS RESEARCH; CHEMICAL AND BIOLOGICAL WARFARE ISSUES; BIOLOGICAL AND TOXIN RESEARCH, HIGH-ENERGY LASERS AND PARTICLE BEAM TECHNOLOGY; ARMS CONTROL TREATY

F. SNYNOPSIS *(Provide a brief description of the system, process, or technology, including its state of development and whether the application is military, commercial, or dual-use).* EXAMPLE: This is new concept of applying current development for high-powered lasers, specifically for military applications.

The paper presents two inventory models for use in commercial remanufacturing operations.

G. REFERENCES *(Check all that apply)*

☒ ALL REFERENCES ARE UNCLASSIFIED, UNLIMITED, AND ARE AVAILABLE TO THE PUBLIC

☐ REFERENCE # ARE SUBJECT TO DISTRIBUTION LIMITATION. NO LIMITED INFORMATION FROM THESE REFERENCES IS INCLUDED IN THE DOCUMENT

☐ NO REFERENCES ARE INCLUDED IN THIS DOCUMENT

RELEASE AUTHORITY AND MANAGEMENT RESPONSIBILITY FOR JOINT EFFORTS

IS THIS INFORMATION CO-AUTHORED WITH SOMEONE FROM AN ORGANIZATION OUTSIDE OF AFIT. ☐ YES ☒ NO	DOES YOUR ORGANIZATION HAVE THE AUTHORITY TO RELEASE THE INFORMATION ☒ YES ☐ NO
IF WORK IS SPONSORED BY ANOTHER ORGANIZATION, DO YOU HAVE THE AUTHORITY TO RELEASE ALL THE INFORMATION ☒ YES ☐ NO	IF NOT, WHAT ORGANIZATION SHOULD COORDINATE ON THE REQUEST ORGANIZATION: PHONE:

NOTE: If the organization is a DOD organization, AFIT/PA will coordinate release approval with that organization's PA office

I. AUTHOR/TECHNICAL REVIEW CERTIFICATION	
AUTHOR	
I certify the information contained in the attached document is technically accurate and does not disclose classified, sensitive, or militarily critical technology and does not violate proprietary rights or copyright restrictions. All security and technology issues listed above have been considered and any applicable security classification guides have been reviewed.	
SIGNATURE	DATE 24 Dec 03
AUTHOR(S) NAME (Print) KEVIN J. GAUDETTE	TELEPHONE (334) 953-0944
OFFICE SYMBOL/ORGANIZATION AF LOGISTICS MANAGEMENT AG.	E-MAIL ADDRESS kevin.gaudette@maxwell.af.mil
TECHNICAL REVIEWER CERTIFICATION (Faculty Advisor)	
I certify the information contained in the attached document is technically accurate and does not disclose classified, sensitive, or militarily critical technology and does not violate proprietary rights or copyright restrictions. All security and technology issues listed above have been considered and any applicable security classification guides have been reviewed.	
SIGNATURE	DATE
TECHNICAL REVIEWER'S NAME (Print)	TELEPHONE
OFFICE SYMBOL/ORGANIZATION	E-MAIL ADDRESS
DEPARTMENT HEAD *(or equivalent)*/PROGRAM MANAGER CERTIFICATION	
The attached material submitted for public release has been properly staffed and reviewed by this division/organization. I support the recommendation of the Author and Technical Reviewer for public release.	
SIGNATURE	DATE 16 Jan 04
NAME *(Print)* Daniel P. Clepper, Maj	TELEPHONE 5-2259 x 3045
OFFICE SYMBOL/ORGANIZATION C16K	
AFIT PUBLIC AFFAIRS	
THE ATTACHED MATERIAL IS/IS NOT CLEARED FOR PUBLIC RELEASE	
SIGNATURE OF SECURITY AND POLICY REVIEW OFFICER Pat Wagner	DATE 21 Jan 04
J. REMARKS *(please reference the specific section)*	

GENERAL INSTRUCTIONS
The information provided on this worksheet will be used by the Air Force Institute of Technology Public Affairs Directorate to process requests for Security and Policy Review in accordance with Air Force Instruction 35-101, Chapter 15. Do not use it to request clearance of software source code or web pages. To expedite processing of your request, please print legibly and complete all sections of the worksheet. Space is provided in Section K for additional remarks. If used, please specify the section commented on. Please submit one copy of both this worksheet and the document. We will contact you if additional copies are required. Documents may be submitted in print or digital format (CD-ROM, DVD, or Zip Disk) or VHS videotape.

SECTION INSTRUCTIONS	
SECTION A:	**DOCUMENT TYPE** - Identifies the intended presentation format for the information. Place a check in the appropriate box.
SECTION B:	**TITLE OF DOCUMENT** - Complete document title *(Explain abbreviations/acronyms in Section J)*
SECTION C:	**AUTHOR(S) NAME, DUTY TITLE, OFFICE SYMBOL** - Author's first and last name, position or duty title and complete office symbol (e.g., AFIT/PA)
SECTION D:	**FORUM** - Indicate the method that will be used to present the information. If for presentation at a conference or meeting, provide the full name of the conference or meeting. Provide the location (city, state) and date of the event. If for print media, provide the name of the publication and submission deadline. If related documents have been previously cleared for public release, provide the AFIT case number. *(An example of a document that has been previously cleared is as follows: If a thesis has been cleared and you are extracting information from that thesis for another article you have written. Please include the case number of the thesis.)*
SECTION E:	**NATIONAL SECURITY AND TECHNOLOGY ISSUES** - Answer each question and explain positive responses in Section F or J.
SECTION F:	**SYNOPSIS** - Provide a brief description of the system, process, or technology, including its state of development and whether its application is military, commercial, or dual-use.
SECTION G:	**REFERENCES** - Check all that apply.
SECTION H:	**JOINT AND INTERNATIONAL EFFORTS** - Indicate whether or not the information resulted from any effort with an external organization or person or with another DOD or U.S. government agency and specify which organization has release authority and program management responsibility.
SECTION I:	**TECHNICAL REVIEW AND CERTIFICATION** - The Author and Technical Reviewer (faculty advisor) must certify that the information complies with the statement. (The technical reviewer section applies only when students are submitting documents.) A department head or deputy (or equivalent) or program manager must sign in the second block affirming support for public release.
SECTION J:	**REMARKS:** Use when additional space is needed to explain a response. Please indicate the section to which the explanation applies. Explain any abbreviations or acronyms.

NOTE:

ALL PERSONAL INFORMATION IN THE FOLLOWING CATEGORIES ABOUT U.S. CITIZENS, DOD EMPLOYEES AND MILITARY PERSONNEL SHOULD NOT BE INCLUDED IN THESES, DISSERTATIONS OR OTHER DOCUMENTS THAT WILL BE POSTED ON THE WORLD-WIDE WEB:

1) SOCIAL SECURITY ACCOUNT NUMBERS
2) DATES OF BIRTH
3) HOME ADDRESSES
4) TELEPHONE NUMBERS OTHER THAN NUMBERS OF DUTY OFFICERS WHICH ARE APPROPRIATELY MADE AVAILABLE TO THE GENERAL PUBLIC
5) NAMES, LOCATIONS AND ANY OTHER IDENTIFYING INFORMATION ABOUT FAMILY MEMBERS OF DOD EMPLOYEES AND MILITARY PERSONNEL

INVENTORY PLANNING FOR REMANUFACTURING

Kevin J. Gaudette

An Abstract of a Dissertation
Submitted to the Faculty of the University Graduate School
in Partial Fulfillment of the Requirements
for the Degree
Doctor of Philosophy
In the Kelley School of Business
Indiana University

December 11, 2003

Vincent A. Mabert

S. Christian Albright

James H. Patterson

Srinagesh Gavirneni

ABSTRACT

Remanufacturing is a process by which a used product called a "core" is restored to like-new condition. The process involves disassembling the core down to its constituent used parts, differentiating between serviceable parts and scrap, and using a combination of serviceable parts and new parts to rebuild the product. A critical component of the process from an operational standpoint is the planning of core purchases, disassemblies, and new part purchases. These three unique inventory planning decisions have been scarcely addressed in the literature to date. This research presents and tests two alternative techniques for solving the remanufacturing inventory planning problem. The first is a deterministic network linear programming formulation with safety stock (ReNet), while the second is a stochastic formulation that explicitly accounts for uncertain yields (SIPR). The experiment is designed to test the relative performance of the two techniques and the effects of yield uncertainty and product characteristics on the solution cost. The results indicate that SIPR outperforms ReNet, particularly for higher levels of uncertainty. Further, product characteristics were shown to have a significant effect on both the relative performance and the solution cost. In particular, the expected yields of the highest-cost parts tend to drive the solution. Remanufacturing managers should therefore focus their resources on increasing the yields of high-cost parts through design and quality control initiatives. Since reducing yield uncertainty also significantly reduces cost, managers should further consider implementing procedures or investing capital to more accurately forecast yields.

INVENTORY PLANNING FOR REMANUFACTURING

Kevin J. Gaudette

Submitted to the Faculty of the University Graduate School
in Partial Fulfillment of the Requirements
for the Degree
Doctor of Philosophy
In the Kelley School of Business
Indiana University

December, 2003

ACCEPTANCE

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements of the Degree of Doctor of Philosophy in Business.

Vincent A. Mabert

Vincent A. Mabert, Chairman

S. Christian Albright

S. Christian Albright

Doctoral Committee

Patterson

James H. Patterson

G. Nagesh

Srinagesh Gavirneni

December 11, 2003

PREFACE

Remanufacturing has been called the "Hidden Giant" in the literature due to its significant, yet relatively silent effect on the economy. In fact in terms of both revenue and employment, it rivals such giants as pharmaceuticals and household consumer durables. Still, since it spans over 45 separate and unique industries, it has received little attention by academicians, making it fertile ground for research. Early on, the unique structure of the remanufacturing inventory planning problem drew my interest.

It has been estimated that over 80% of the life cycle cost of any product is incurred after sale. As such, life cycle-centric products such as those that are remanufactured must pay particularly close attention to areas like logistics and inventory management. Since remanufacturing is likely to play a growing role in the economy in the future, it is critical that these operational areas be addressed by the academic community in order to make remanufacturing economically viable.

I undertook this research to begin to fill this gap. Specifically, I focused on the related problems of determining the optimal number of used products, or "cores", to procure and disassemble and the optimal quantities of new parts to procure. This led to a definition of the problem structure and the formulation of two alternative techniques to solve it. In the process, I gleaned insights into the effects of the uncertainty of part yields from cores and the structural characteristics of products on cost. These insights should begin to help remanufacturing managers make both strategic and operational decisions in the course of their business.

It is my hope that this research will spur additional efforts to help make remanufacturing an economically viable means of meeting product demand in the future. If this end is realized, the staggering environmental benefits can be realized without depending too heavily on the legislative mandates that are becoming popular in Europe and, more recently, in parts of the United States.

Kevin J. Gaudette
Bloomington, Indiana
December 2003

ABSTRACT

Remanufacturing is a process by which a used product called a "core" is restored to like-new condition. The process involves disassembling the core down to its constituent used parts, differentiating between serviceable parts and scrap, and using a combination of serviceable parts and new parts to rebuild the product. A critical component of the process from an operational standpoint is the planning of core purchases, disassemblies, and new part purchases. These three unique inventory planning decisions have been scarcely addressed in the literature to date. This research presents and tests two alternative techniques for solving the remanufacturing inventory planning problem. The first is a deterministic network linear programming formulation with safety stock (ReNet), while the second is a stochastic formulation that explicitly accounts for uncertain yields (SIPR). The experiment is designed to test the relative performance of the two techniques and the effects of yield uncertainty and product characteristics on the solution cost. The results indicate that SIPR outperforms ReNet, particularly for higher levels of uncertainty. Further, product characteristics were shown to have a significant effect on both the relative performance and the solution cost. In particular, the expected yields of the highest-cost parts tend to drive the solution. Remanufacturing managers should therefore focus their resources on increasing the yields of high-cost parts through design and quality control initiatives. Since reducing yield uncertainty also significantly reduces cost, managers should further consider implementing procedures or investing capital to more accurately forecast yields.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Remanufacturing is a process in which a used product called a "core" is restored to like-new condition. The process involves disassembling the core down to its constituent used parts, differentiating between serviceable parts and scrap, and using a combination of serviceable parts and new parts to rebuild the product. Many firms, among them General Electric, Caterpillar, Lockheed Martin, Pitney Bowes, and Cummins Engine, have recognized that there are significant business opportunities in the aftermarket for remanufactured goods. In fact, an estimated 73,000 firms in the U.S. are involved in remanufacturing, accounting for over $53 billion in sales and employing a half-million people (Lund 1996). In terms of sales and employment, the remanufacturing "industry" rivals the household consumer durables, steel mill products, computers and peripherals, and pharmaceuticals industries.

Growth in remanufacturing is being driven by two primary factors. First is its increased acceptance by both firms and consumers, the latter capitalizing on the lower cost of remanufactured goods and the former on a stable source of profit (Giuntini and Gaudette 2003). Since the seminal works by Lund (1984, 1996), researchers and trade groups have pooled resources to offer strategic guidance and research to remanufacturers (Thierry et al. 1995; Hormozi 1997), as well as to promote its benefits to lawmakers. The second driver is its perceived positive impact on the environment. Several researchers

have made strong cases for the environmental benefits of remanufacturing (e.g. Ayres et al. 1997; O'Brien 1999; and Giuntini and Gaudette 2003), and lawmakers in Europe and the United States have answered the call by passing legislation designed to encourage its use. In order for remanufacturing to become an economically viable alternative to manufacturing, however, many operational issues need to be addressed and researched.

Although remanufacturing shares some operational characteristics with the traditional manufacturing process (shop floor scheduling and assembly, e.g.), it differs in several important respects. First, certain elements of the supply chain are unique, such as the reverse logistics network to reclaim cores. Cores are procured directly from customers via exchange programs, indirectly through distributors or retailers, and/or from third-party brokers at market prices (Figure 1-1). Second, remanufacturing requires special operational processes and skills, such as disassembly, inspection, testing, and repair. And third, inventory planning decisions unique to the remanufacturing environment must be made regularly on a rolling planning horizon. Such decisions include the number of cores to procure, the number of cores to disassemble in each period, and the number of new parts needed to replace scrapped parts. These are challenging decisions, in light of the high level of uncertainty inherent in the remanufacturing process and the unique, integrated structure of the inventory planning decision. This research focuses on the latter, developing and testing two inventory planning techniques that address the unique uncertainties and decisions faced by remanufacturers.

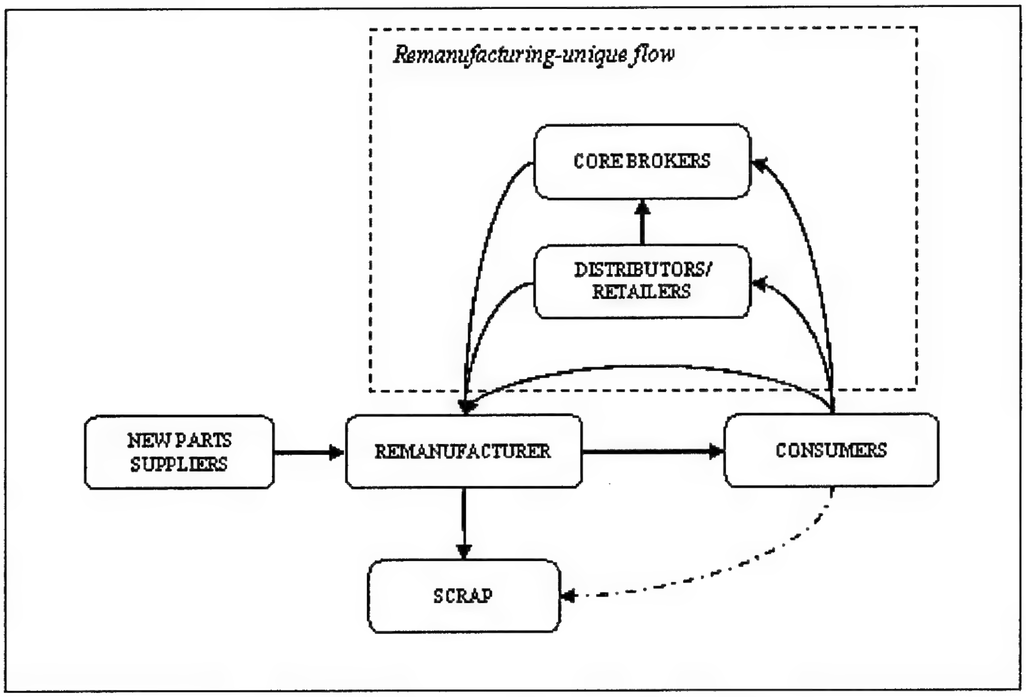


Figure 1-1: The Remanufacturing Supply Chain

Inventory Planning in Remanufacturing

Remanufacturers and manufacturers face many of the same inventory challenges. End-item demand uncertainty, supply uncertainty, and lead time uncertainty, for example, are problematic in both environments. However, inventory planning in remanufacturing is more challenging than that of its manufacturing counterpart for two primary reasons. First, the levels of uncertainty faced by remanufacturing inventory planners are significantly higher. Specifically, supply and demand uncertainties are much more problematic in remanufacturing due to the uncertain nature of the product failures that drive core supply and end-item demand. Even more problematic is the uncertain yield of usable parts that emanates from the cores upon disassembly. For each core disassembled,

each part p will be usable with a probability γ_p and scrapped with a probability $(1-\gamma_p)$. In aggregate, the *yield* of usable parts (ψ_p) is a product of the number of cores disassembled (Q^d) and the yield percentage (γ_p). To simplify the initial discussion, the variables in equation (1) are assumed to be deterministic. However, later discussion relaxes this assumption and treats ψ_p and γ_p as random variables with known distributions.

$$\psi_p = Q^d \gamma_p \tag{1}$$

To illustrate, consider the simple example of a single-use camera. For simplicity, assume that there are two potentially reusable parts in the camera: the plastic shell (part 1) and the microchip (part 2). Further assume that the plastic shell is reusable with a probability of $\gamma_1 = 0.9$ and that the microchip is reusable with a probability of $\gamma_2 = 0.7$. For this simple example, if 100 cores are disassembled (i.e. $Q^d = 100$) the expected yields of parts 1 and 2 are $\psi_1 = 90$ and $\psi_2 = 70$, respectively. Although most remanufacturers assume the yield to be deterministic to simplify planning, in practice it is highly uncertain and also varies from part to part. In combination supply, demand, and yield uncertainty make remanufacturing a very different and more challenging environment than manufacturing from the perspective of inventory planning. Lead time uncertainty, while not unique to remanufacturing, can exacerbate the problems. Figure 1-2 identifies the three sources of uncertainty discussed above with dotted lines, while the dashed lines illustrate the data flow from these uncertain elements that must be accounted for in the forecast of new part requirements.

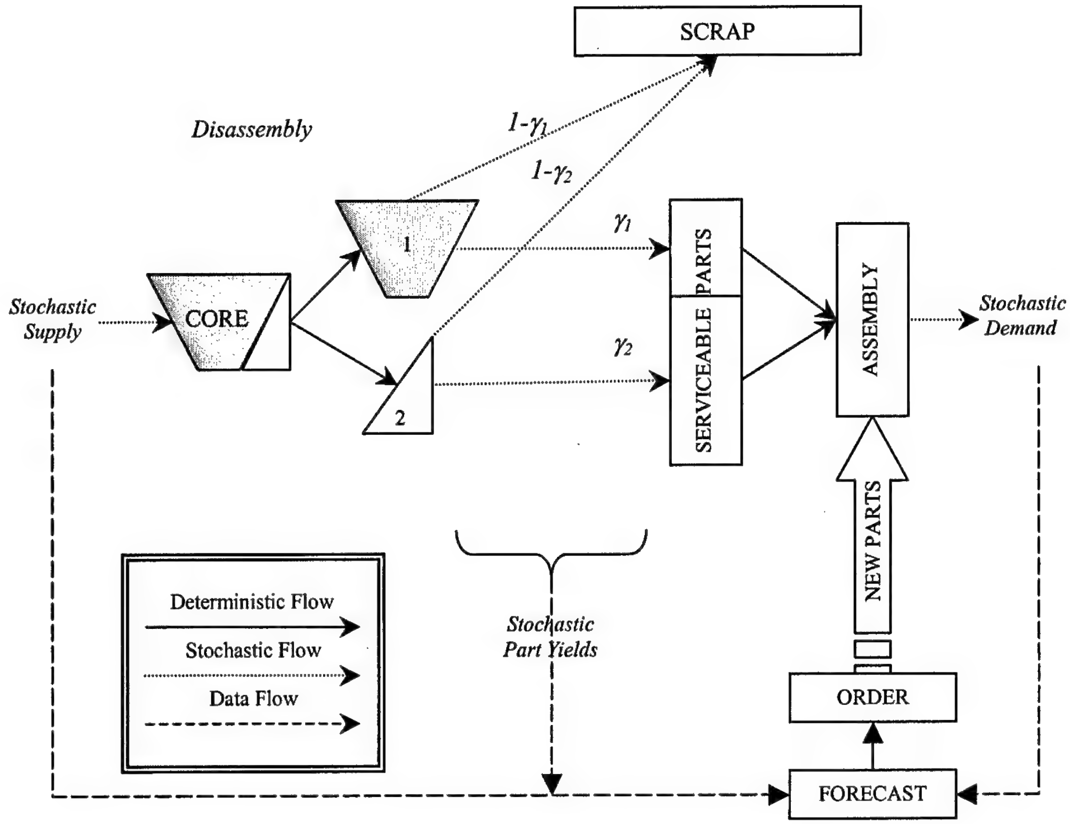


Figure 1-2: Material and data flows in the remanufacturing process

The second major difference with regard to remanufacturing inventory planning is the structure of the inventory decision itself. In manufacturing, new component requirements are generally calculated directly via an MRP explosion based on the end-item requirements in the master production schedule. The components are then ordered from suppliers using a lead time offset. For the camera example, 100 microchips and 100 plastic shells are required to manufacture 100 cameras. The parts are ordered at a point in time so that the orders arrive just in time for assembly, with safety lead time added to account for lead time uncertainty and safety stock to account for quantity uncertainty, if necessary.

In remanufacturing, by contrast, two types of supply sources exist. The primary source for parts is used cores, each of which yields an uncertain set of usable components. The secondary (and generally more expensive) source is the traditional new parts supply chain, which experiences much lower quantity uncertainty and higher lead time uncertainty. But since the quantity of cores disassembled affects the requirements for new parts, inventory planning involves an integrated, two-dimensional decision. The additional consideration of cores and used part yields, highlighted in Figure 1-3, makes inventory planning different from that of traditional manufacturing and uniquely complex.

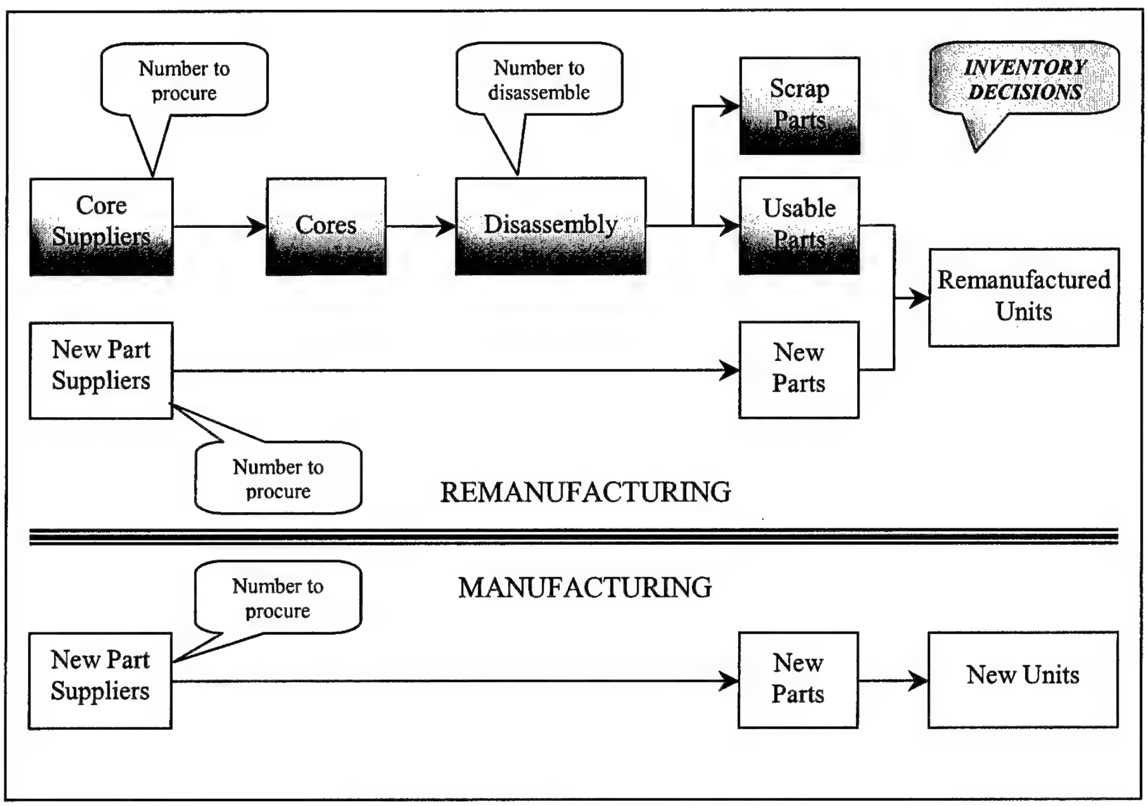


Figure 1-3: Remanufacturing and Manufacturing material flows and planning decisions

To illustrate the integrated nature of the planning decision that will be discussed for the remainder of the dissertation, the following parameters are assumed for the camera example. For simplicity, yields are assumed to be deterministic and 200 cores are currently in stock.

Table 1-1: Problem Parameters

Part	Yield %	New Part Cost
Plastic Shell	0.9	$10
Microchip	0.7	$4
Core Cost = $4 Disassembly Cost = $2 Demand = 100 units Annual holding cost = 15%		

Beginning with an extreme case, if no cores are disassembled then the net yield of used parts is likewise zero for both parts. In this case, to meet the demand of 100 a full set of 100 new parts must be purchased. The cost of $1400 reflects only the cost of purchasing 100 units of each part (this case is equivalent to manufacturing 100 new units). As more cores are disassembled, increasing quantities of used parts become available, reducing the number of new parts needed to meet demand. At the same time, increasing costs are incurred for purchasing and disassembling cores and, at some point, for holding or disposing of excess used parts in inventory. Note that the new part requirements depend on the quantity of cores disassembled, which illustrates the integrated and unique nature of the planning problem. Although the calculations are trivial for this simple illustrative example, the case with uncertain yields becomes extremely complex due to the integrated nature of the decision and the size of the solution space. The remaining part requirements are shown for increasing numbers of

cores disassembled in Table 1-2, with the optimal solution (111 cores disassembled) highlighted in gray.

Table 1-2: Example Part Requirements and Costs

Cores Disassembled	Part 1		Part 2		Purchase Cost	Excess Cost	Core & Disassembly Costs	Total Cost
	Purchase	*Excess*	*Purchase*	*Excess*				
0	100	0	100	0	1400	0.00	0	1400.00
•								
•								
105	6	0	27	0	168	0.00	630	798.00
106	5	0	26	0	154	0.00	636	790.00
107	4	0	25	0	140	0.00	642	782.00
108	3	0	24	0	126	0.00	648	774.00
109	2	0	24	0	116	0.00	654	770.00
110	1	0	23	0	102	0.00	660	762.00
111	0	0	22	0	88	0.00	666	754.00
112	0	1	22	0	88	0.03	672	760.03
113	0	2	21	0	84	0.06	678	762.06
114	0	3	20	0	80	0.09	684	764.09

Planning Model Requirements

The discussion to this point has identified two fundamental requirements for an effective remanufacturing inventory planning model. First, it must be designed to solve two seemingly separate inventory planning problems in an integrated manner. These are the model decision variables later used in Chapter 3.

- Quantity of cores to disassemble (Q^d)
- Quantity of each part to buy new (Q_p, $p = 1 \ldots P$)

The second requirement is that it account for uncertainty. As previously noted, supply, demand, yield, and lead time uncertainties all exist in the remanufacturing environment. For the purposes of this research, the scope is limited to yield uncertainty since it is the

most unique to remanufacturing. Supply uncertainty (for new parts), end-item demand uncertainty, and lead time uncertainty have all been studied extensively in the literature. As such, they are assumed here to be deterministic in order to isolate the effects of yield uncertainty without confounding effects.

Two inventory planning techniques are developed in Chapter 3 that are later compared in a set of analytical experiments. They follow the same general single-period formulation, and are designed to address the two requirements identified above.

$$\text{MIN} \quad (c_c + c_d)Q^d + \sum_p c_p Q_p + \sum_p (I_p + Q_p + E[\Psi_p] - D_p)^+ c_{hp} \tag{2}$$

$$\text{ST} \quad \Pr[(I_p + E[\Psi_p] + Q_p) \geq D_p] \geq TSL \tag{3}$$

$$Q^d \leq C_d \tag{4}$$

$$Q^d \leq S_c \tag{5}$$

$$D^u \leq C_a \tag{6}$$

$$Q^d, Q_p \qquad \text{Positive integers} \tag{7}$$

Where:
- c_c = Core purchase cost
- c_d = Disassembly cost
- c_p = Purchase cost of part *p*
- c_{hp} = Holding cost of part *p*
- Q^d = Quantity of cores disassembled
- Q_p = Purchase quantity of part *p*
- I_p = Starting inventory of part *p*
- D_p = Demand for part *p*
- Ψ_p = Yield (units) of part *p*
- γ_p = Yield percentage of part *p*
- TSL = Target service level
- D^u = End-Item demand (units)
- C_d = Disassembly capacity
- C_a = Assembly capacity
- S_c = Available core supply

The objective function (2) minimizes the sum of the core purchase and disassembly costs, new part purchase costs, and excess holding/disposal costs. Constraint (3) ensures that the end-item service level, defined as the probability of having enough complete sets of parts to meet demand, meets or exceeds some target service level (TSL). Constraints (4) and (6) enforce the disassembly and assembly capacities of the plant, respectively, while constraint (5) enforces the supply constraint on cores. Constraint (7) ensures all decision variables are positive integers.

The first solution technique initially assumes that yields are deterministic, thus reducing the problem to one that can be optimally solved using a network linear programming approach. The optimal disassembly and part order quantities (Q^d and Q_p, respectively) are then adjusted using modified safety stock techniques to reach a target service level. This approach has the advantages of being intuitive and flexible, and can be easily modified to include multiple products and multiple time periods. For the single-period, single-product case investigated here, its solution is the equivalent of a "reverse bill of material" explosion, a technique commonly used in practice. It also has a limitation in that the calculation of safety quantities is sub-optimal and requires an additional step beyond the network LP solution.

The second approach explicitly includes stochastic yields, which allows a near-optimal solution (for the single period case) to be derived. The stochastic approach is limited in that it becomes intractable for multi-period and multi-product problems, requiring heuristic solutions that might deviate from optimality. The two approaches are compared using a numerical experiment for the single-period, single-product case. A secondary experiment explores their relative performance for the multi-period case.

Research Overview

This research offers three primary contributions that distinguish it from the existing remanufacturing literature. First, it develops two alternative inventory planning techniques, containing the unique requirements described in this section, to help remanufacturers make efficient part-level inventory planning decisions. The objective of both techniques is to minimize the cost of meeting a target service level. Second, a numerical experiment is performed to offer managerial insight into the performance of the two approaches under different levels of yield uncertainty. Since a variety of techniques and technologies exist to reduce the uncertainty of yields, each with an associated cost, quantifying the effects of such reductions becomes a critical factor in deciding whether to pursue them. Finally, a set of experimental problems is created to test the inventory techniques under a wide variety of product structures. This problem set, in addition to aiding with the current analysis, can be used as a baseline for future comparative studies. Specifically, the experiment is designed to answer the following three research questions:

1. To what extent, and under what conditions, does a planning approach that accounts for stochastic yields outperform a deterministic (network LP) approach with respect to cost?
2. What effect does the reduction of yield uncertainty (i.e. prior knowledge of expected yields) have on service level and cost?
3. How do a product's structural characteristics (yield percentages, part costs, and number of parts, e.g.) affect the cost of the solution?

The remainder of the dissertation is organized as follows. Chapter 2 presents a review of the literature, beginning with a general overview of remanufacturing research and a more detailed discussion of remanufacturing inventory studies, and concluding with a comparison between this research and the existing remanufacturing inventory literature. In Chapter 3, two alternative inventory planning models are described using an example problem. The first assumes deterministic yields and is formulated as a network flow linear programming problem, followed by the addition of traditional safety quantities to reach a target service level. The second is a stochastic approach in which the probability distributions of the part yields are explicitly modeled, in effect integrating the order and safety quantity decisions.

Chapter 4 provides the details of the main experiment designed to answer the primary research questions. The relative performance of the two approaches of Chapter 3 will be compared under varying levels of yield uncertainty, using a number of problems designed to illustrate the pertinent characteristics that drive the solution cost. Also included in Chapter 4 is a more detailed discussion of the experimental factors, levels, and performance measures. Chapter 5 presents the results of the main experiment and a discussion of their implications, and is organized according to the research questions. Chapter 6 presents the results of the sensitivity analysis and secondary experiment, both of which explore the research questions in more detail. Chapter 7 offers concluding remarks and suggestions for future research in the area.

CHAPTER 2

LITERATURE REVIEW

Despite the size and scope of remanufacturing activity in the U.S., it has been given relatively little attention in terms of academic research. This is at least partially due to the fact that remanufacturing takes place in some 85 different industries, each with its own unique characteristics, making generalizable research difficult. The published research can be divided into three broad areas, highlighted below.

1. Production Planning and Control

Over the past decade, several authors have focused their efforts on identifying production issues unique to remanufacturing. Guide, Srivastava, and Spencer (1996), for example, tested existing rough-cut capacity planning (RCCP) techniques and determined that they do not adequately address the complexities found in most remanufacturing environments. Subsequent work began to fill this gap by developing scheduling policies (Guide, Kraus, and Srivastava 1997), dispatch rules (Guide 1997), and expediting policies (Guide, Srivastava, and Kraus 1998) for remanufacturing. Guide (2000) also presents an extensive literature review for remanufacturing production planning and control and identifies unique research issues that still need to be addressed. Finally, Souza, Ketzenberg, and Guide (2002) formulate a queuing network to solve the product mix decision with service level constraints.

2. Core Forecasting

While most research has assumed that core returns are independent of demand, i.e. the system is open, several researchers have attempted to model the special case where returns and demands are correlated. Kelle and Silver (1989) published a seminal work in this area, focusing on returnable containers. Toktay, Wein, and Zenios (2000) modeled demands and returns as a closed-queuing network and developed a Bayesian procedure to estimate and dynamically update return probabilities and lag times. Most recently, Guide and Van Wassenhove (2001) proposed an approach to proactively manage the timing, quality, and cost of returns.

3. Inventory Planning and Control

Inventory research has dominated the remanufacturing literature over the past decade. Since it is also the topic of this dissertation, a more detailed review is presented to help position this research. Table 2-1 summarizes the taxonomy of the remanufacturing inventory literature in terms of several characteristics: type (deterministic or stochastic), scope (remanufacturing only or manufacturing/ remanufacturing *hybrid*), core return assumption (closed or open system), and level of analysis (end-item- or component-level focus). It also includes a brief description of the types of decisions addressed in each case.

Deterministic Inventory Models

The top portion of Table 2-1 catalogs six research efforts that assume deterministic conditions. The first four further assume a hybrid system in which both

Table 2-1: Taxonomy of Remanufacturing Inventory Literature

Author(s)	Year	Type	Scope	Open/ Closed	Level of Analysis	Types of decisions
Richter & Dobos	1999	Deterministic	Hybrid	Open	End-item	Repair and disposal quantities using EOQ-based model
Richter & Weber	2001			Open	End-item	Remanufacturing and manufacturing batch timing using Reverse Wagner-Whitin
Golany, Yang, & Yu	2001			Open	End-item	Remanufacturing, manufacturing, and disposal decisions per period
Teunter & van der Laan	2002			Closed	End-item	Non-optimality of average cost approach
Jayaraman, Guide, & Srivastava	1999		Reman.	Closed	End-item	Shipping quantities and facility locations
Ferrer & Whybark	2001			Closed	Components	Core procurement and disassembly and new part ordering
van der Laan, Salomon, Dekker, & van Wassenhove	1999	Stochastic	Hybrid	Open	End-item	Remanufacturing and manufacturing batch timing
Teunter, van der Laan, & Inderfurth	2000			Closed	End-item	Alternative methods for setting holding costs
Toktay, Wein, & Zenios	2000			Closed	Components*	Order quantities
Inderfurth & van der Laan	2001			Open	End-item	Remanufacturing and manufacturing batch timing with different lead times
Kiesmuller & van der Laan	2001			Closed	End-item	Remanufacturing and Manufacturing Batch Timing
Teunter & Vlachos	2002			Open	End-item	Necessity of a disposal option using simulation
Muckstadt & Isaac	1981		Reman.	Open	End-item	Remanufacturing & outside procurement decisions
van der Laan, Dekker, & Salomon	1996			Open	End-item	Remanufacturing, disposal, & outside procurement decisions
Guide & Srivastava	1997			Open	Components	Safety stock levels

* Single component with 100% yield, equivalent to end-item level of analysis

manufacturing and remanufacturing operations exist together to meet the same end-item demand. Richter and Dobos (1999) use an analytic model based on Economic Order Quantity (EOQ) logic to determine the optimal lot quantities of end-items for repair and disposal. Richter and Weber (2001) apply the Reverse Wagner-Whitin algorithm to optimally time manufacturing and remanufacturing batches of end-items. Golany, Yang, and Yu (2001) focus on a similar problem to that of Richter and Weber (2001), but add the core disposal decision as well. They use a network linear programming formulation

to generate a solution. Teunter and van der Laan (2002) offer support for using discounted cash flows (DCF) in lieu of the more commonly used average cost (AC) approach in remanufacturing inventory models.

The remaining two deterministic efforts focus on remanufacturing activities that are independent of manufacturing, both presenting inventory planning approaches. Jayaraman, Guide, and Srivastava (1999) present a network model that solves the end-item distribution and facility location problems simultaneously. Ferrer and Whybark (2001) present an approach that differs significantly in that it focuses on planning decisions at the component level instead of the end-item level. They present a pair of unconstrained linear programming models, one to determine the minimum number of cores to order at the start of the planning horizon and a second to calculate the minimum number of cores to disassemble in each period, in order to ensure that part requirements are met in all periods. The model also accounts for component commonality across multiple product types.

Stochastic Inventory Models

The bottom half of Table 2-1 presents those research efforts that account for stochastic elements. Early work (Muckstadt and Isaac, 1981; van der Laan, Dekker, and Salomon, 1996) focused on independent remanufacturing operations and solved the remanufacturing, outside procurement, and disposal decisions at the end-item level. Much of the later work shifted to the formulation and solution of remanufacturing and manufacturing batch timing decisions in a hybrid system (van der Laan et al., 1999; Inderfurth and van der Laan, 2001; and Kiesmuller and van der Laan, 2001). Several

others have looked at specific remanufacturing inventory issues, such as holding costs (Teunter, van der Laan, & Inderfurth 2000), the necessity of a disposal option for excess cores (Teunter & Vlachos 2002), and safety stocks to buffer against uncertainty (Guide & Srivastava 1997). Finally, Toktay et al. (2000) developed a closed-queuing network model to estimate core returns.

Of the stochastic research outlined above, only Guide and Srivastava (1997) directly address part-level planning decisions. Toktay, Wein, and Zenios (2000), determine part requirements, but single out one high-cost part and assume a 100% yield of that part from all returned cores. The only stochastic element explicitly modeled is the core return itself, effectively making the objective equivalent to those that focused on end-item inventory control. Their approach entails a closed queuing network model which uses a Bayesian technique to estimate returns based on past sales.

Position of Research

To position the current research, the existing literature is more narrowly categorized in Figure 2-1 according to two of the defining characteristics from Table 2-1. The first involves the level of analysis of each model, and therefore its objective. The vast majority of the literature has concentrated on end-item decisions, presenting models that determine end-item inventory levels at which manufacturing and remanufacturing batches should begin, for example. The research reported here, by contrast, has a *part-level* planning focus. The second characteristic is the inclusion (or exclusion) of stochastic elements in the model. In the case of end-item models, this implies stochastic

core supply or end-item demand, while for part-level models it implies stochastic yield rates for individual parts from disassembled cores.

Implicit in the end-item approach are several assumptions that warrant discussion. First, since both manufacturing and remanufacturing decisions are being made, the systems are assumed to include a hybrid of both operations. Second, and more importantly, all models in this class assume that there is no market difference between new and remanufactured products. In other words, either can be used to meet the same market demand at the same market price. And third, these models do not address the part-level inventory planning decisions regularly faced by remanufacturers.

Core Supply/Part Yield Assumption	***Component Part***	***End-Item***
Stochastic	Guide & Srivastava (1997) ***Proposed Research***	Muckstadt & Isaac (1981) van der Laan et al. (1996) van der Laan et al. (1999) Teunter et al. (2000) Toktay, Wein, & Zenios (2000) Inderfurth & van der Laan (2001) Kiesmuller & van der Laan (2001) Teunter & Vlachos (2002)
Deterministic	Ferrer & Whybark (2001)	Richter & Dobos (1999) Jayaraman et al. (1999) Richter & Weber (2001) Golany, Yang, & Yu (2001) Teunter & van der Laan (2001)

Level of Analysis

Figure 2-1: Position of Proposed Research

The limitations cited above by no means detract from the relevance of the literature, since many examples certainly exist of original equipment manufacturers (OEMs) that produce simple products with few replaceable parts and also engage in remanufacturing (the Kodak example from Toktay et al. 2000, e.g.). As such, the assumption of a hybrid system and the focus on end-item inventory decisions are valid in some settings. Still, many more examples exist in which remanufacturing stands as an independent operation, either as a separate business unit within an OEM's structure or as the sole business of an independent (non-OEM) remanufacturer. In fact, as Lund (1996) points out, the majority of remanufacturers in the U.S. are non-OEMs with fewer than 6 employees generating less than $400,000 in sales. Furthermore, the market demands and prices of new and remanufactured goods are typically quite different and distinct. And finally, most remanufactured goods contain multiple reusable parts, each with a distinct yield distribution, that must be accounted for when planning and ordering inventory. This research therefore assumes remanufacturing to be an independent business unit with unique production, demand, price, and cost characteristics.

The second defining characteristic of this research is the inclusion of stochastic yields. As illustrated in Figure 2-1, very little published research has focused on part-level planning decisions. Of the two works cited in this category, Guide and Srivastava (1997) are the only researchers to date who have addressed the uncertainty of part yields and its effect on part-level planning. Specifically, they tested the use of traditional safety levels for individual parts and offered insights into their effectiveness in a remanufacturing environment.

The approaches developed in the following chapter account for stochastic part yields and solve the core disassembly and new part ordering problems simultaneously. Unlike Toktay, et al. (2000), they assume an open system in which returns are not directly linked to sales, which implies a relatively long and variable product life prior to remanufacturing. They also address varying part yields across multiple parts, further differentiating them from Toktay et al. (2000).

The stochastic model draws upon the work of Feeney & Sherbrooke (1966) and Sherbrooke (1971), which were originally developed for a repair setting. Although repair systems have been thoroughly studied throughout the years, remanufacturing can be distinguished in three ways. First, repair systems are steady-state, closed systems in which the total number of end-items remains constant, while in remanufacturing the population of end-items is constantly changing. Second, because of the nature of repair, systems generally determine steady-state stock levels to ensure that needed repairs can be accomplished, while in remanufacturing parts are ordered on a periodic basis to meet forecasted production requirements. And finally, repair systems typically assume that a single part failure will drive the necessity for a repair. In remanufacturing, by contrast, the possibility exists that each individual part may need to be replaced.

Feeney & Sherbrooke (1966) present a single-site repair cycle model that sets inventory stock levels for each part. The objective of the model is to minimize the expected backorders subject to a cost constraint. Sherbrooke (1971) extends this approach and proves its mathematical equivalence to an objective of minimizing the cost of reaching a target system availability. Although the structure of the problem and the

definition of availability are different in this research, the greedy algorithm they use to find an efficient solution quickly forms the basis of the stochastic approach reported here.

Finally, a word on random yield literature is warranted since, on the surface, it appears to parallel this work. The random yield literature was categorized by Gurnani, Akella, and Lehoczky (2000) into three areas: (1) the random yield problem; (2) supplier diversification; and (3) the assembly problem. The first set attempts to optimize individual lot sizing decisions when random yields are present, such as in the case of electronic components like computer chips, a problem quite different from the remanufacturing problem. The second set addresses the problem of multiple suppliers with different yield means and variances. The latter has commonly been compared to investment diversification literature, where the mean yield is analogous to the expected return and the yield variance to investment risk. The assembly problem is structurally the most similar to the remanufacturing problem addressed here, in that it solves the production and order quantity decisions simultaneously. The main distinction between the two is that in the remanufacturing problem, the disassembly quantity affects each component in a different way. By contrast, in the assembly problem the production quantity decision affects all order quantities equivalently. The remanufacturing problem therefore requires that the disassembly quantity and order quantities be solved in an integrated manner, as will be discussed in more detail in Chapter 3.

CHAPTER 3

INVENTORY PLANNING TECHNIQUES FOR REMANUFACTURING

In a broad sense, the remanufacturing problem involves a set of integrated inventory planning decisions across a rolling planning horizon. Since many remanufacturing firms use Material Requirements Planning (MRP) or similar systems for inventory planning, it is modeled here as a periodic review system, similar in function to the net requirement calculations of a capacitated MRP system. In this capacity, at discrete points in time the quantity of cores to disassemble (Q^d) and the net requirements for new parts (Q_p, $p = 1 \ldots P$) are determined for each future period in the planning horizon. Actual orders for cores and parts are then scheduled using a lead-time offset, as in MRP.

What makes the problem unique are (1) the uncertainty of part yields from the core disassembly process and (2) the integrated nature of the decision variables Q^d and Q_p. This chapter discusses the mathematical structure and complexity of the problem in more detail and presents the details of two alternative methodologies for solving it. The methodologies are developed for the general case, to include multiple periods and cores, although the main experiment focuses on the single-period, single-core case in order to fully examine the properties of the problem.

Problem Formulation

The formulation of the problem is given by equations (2) through (7) from Chapter 1, repeated below in general form with the addition of multiple periods and cores.

$$\text{MIN} \quad \sum_i \sum_t (c_{ci} + c_{di}) Q_{it}^d + \sum_p \sum_t c_{pt} Q_{pt} + \sum_p \sum_t E[excess]_{pt} c_{hp} \tag{8}$$

$$\text{ST} \quad \prod_p \Pr[(I_{pt} + \Psi_{pt} + Q_{pt}) \geq D_{pt}] \geq TSL \qquad \forall t \tag{9}$$

$$\sum_i Q_{it}^d \leq C^d \qquad \forall t \tag{10}$$

$$Q_{it}^d \leq S_t^c \qquad \forall t \tag{11}$$

$$\sum_i D_{it}^u \leq C^a \qquad \forall t \tag{12}$$

$$Q_{it}^d, Q_{pt} \geq 0, \text{int} \qquad \forall i,t; \forall p,t \tag{13}$$

Where:

c_{ci}	= Core i purchase cost
c_{di}	= Core i disassembly cost
c_{pt}	= Purchase cost of part p, period t
C_{hp}	= Single-Period holding cost of part p
$Q^d{}_{it}$	= Quantity of cores i disassembled in period t
Q_{pt}	= Purchase quantity of part p, period t
I_{pt}	= Starting inventory of part p, period t
D_{pt}	= Demand for part p, period t
Ψ_{pt}	= Yield (units) of part p, period t
γ_p	= Yield percentage of part p
TSL	= Target service level
$D^u{}_{it}$	= End-Item demand (units) for product i, period t
C^d	= Disassembly capacity
C^a	= Assembly capacity
$S^c{}_{it}$	= Available supply of core i, period t

The objective function (8) minimizes the sum of core, part, and holding/disposal costs, where Q^d and the set $\{Q_p\}$ are the decision variables. The first two terms are

straightforward in that they are directly calculated for a given Q^d and set $\{Q_p\}$. The third, representing the cost of holding and/or disposing of excess inventory, warrants further explanation because it is stochastic in nature. The expected excess of each part must account for the probability distribution associated with only those cases where the sum of the yield (a random variable), starting inventory I_p, and order quantity Q_p are greater than the demand D_p. For each part, its expected excess is given by (14) below. The limits of the summation ensure that only the range between an excess of 1 unit and the maximum excess of ($Q^d + I_p + Q_p - D_p$) are included in the calculation.

$$E[excess]_p = \sum_{x=D_p-I_p-Q_p}^{Q^d} (x + I_p + Q_p - D_p)\Pr(X = x) \tag{14}$$

The expected holding/disposal cost for each part p is the product of its single-period holding/disposal cost and its expected excess, as given in (15) and inside the summation of (8).

$$E[holding\_\cos t]_p = E[excess]_p * c_{hp} \tag{15}$$

Constraints (10) through (12) are straightforward capacity constraints, and are therefore omitted from the remainder of the discussion for brevity. Constraint (9) is now discussed in greater detail, since it is the key to accounting for stochastic yields. The left hand side of constraint (9) is the end-item service level. From an inventory perspective, it represents the availability of component parts. In other words, it is the probability that enough complete sets of parts are available to assemble the number of units demanded in a period. The quantity of each part p available in a period t is the sum of its starting

inventory I_{pt}, new parts purchased Q_{pt}, and the yield from disassembled cores Ψ_{pt}. Since the latter is a random variable, the available quantity for each part is also a random variable for which a service level can be calculated. The end-item service level is then the product of the service levels of the component parts.

Before discussing solution methodologies, the mathematical complexity of the problem is briefly described in the following section. To simplify the discussion, the remainder of the chapter assumes the single-period problem used in the main experiment.

Mathematical Complexity

Even for the single-period case, the problem size is prohibitively large, since explicitly modeling stochastic yields results in a very large state space. Recall that there are two decision variables (actually, one is a set) in the problem: the quantity of cores to disassemble (Q^d) and the quantity of each part to purchase (Q_p, $p = 1, 2, \ldots, P$). The number of possible solutions is therefore equal to the number of feasible disassembly quantities multiplied by the product of the numbers of feasible order quantities for all parts. The range of feasible disassembly quantities is bounded on the lower end by 1, since by definition remanufacturing involves disassembling used cores. At the high end, the range is bounded by the minimum of the disassembly capacity of the remanufacturer, C^d (constraint 10) and the number of cores available, S^c (constraint 11).

$$\text{Range } (Q^d) = [1, \min\{S^c, C^d\}] \tag{16}$$

The feasible order quantities for each part range from 0 to the part demand minus any starting inventory.

$$\text{Range } (Q_p) = [0, (D_p - I_p)] \tag{17}$$

The problem size is given by the product of (16) and (17).

$$\text{Problem Size} = [\min\{S^c, C^d\}] * \prod_p (D_p - I_p + 1) \tag{18}$$

Even for a simple product with 5 parts, a demand of 10 units, no starting inventory, and a disassembly capacity of 20, there are over 3.2 million potential solutions. Adding one part to the same scenario results in over 35 billion. Clearly, there is a need for either (1) simplifying assumptions to reduce the problem size or (2) exploitation of the structure of the problem to develop an efficient solution methodology. Both options are explored in the remainder of the chapter, which presents the details of two solution methodologies, one using a simplifying assumption (called "ReNet") and one using an efficient solution methodology (called "SIPR").

To illustrate the solution methodologies the following example is introduced, which will be used for the remainder of the chapter. The characteristics of the example problem, while scaled down for simplification, are based on an actual product. The end-item is comprised of nine parts, each with a new cost, quantity per unit, and average yield, as shown in Table 3-1. End-item demand is 50 units and initial inventory is zero for all items.

Table 3-1: Example Problem Data

Core Cost (c_c)	5
Disassembly Cost (c_d)	15
Holding Cost (c_h)	15% per year

Part (p)	Quantity per assembly (z_p)	Average Yield % (γ_p)	New Part Cost (c_p)	Part Demand (D_p)
1	1	0.9	50.00	50
2	1	0.7	30.00	50
3	1	0.6	8.00	50
4	1	0.9	4.50	50
5	1	0.5	3.50	50
6	1	0.3	3.00	50
7	1	0.9	0.50	50
8	1	0.5	0.35	50
9	2	0.3	0.15	100
		Total Cost	$100.00	

ReNet: Remanufacturing Network LP with Buffers

The first approach to solving the problem is a deterministic network linear programming formulation, hereafter called ReNet, with additional buffer order quantities used to satisfy (9) and incorporate the uncertainty of yields. The approach involves two steps. First, the problem is reduced to its deterministic equivalent by the assumption of known yields and solved using a network simplex algorithm. And second, part order quantities are increased by safety levels to achieve the target end-item service level (TSL). The two steps are discussed in more detail below.

Step One: Network Linear Programming Formulation

ReNet involves an initial simplifying assumption that the yields Ψ_p are deterministic (expressed as Y_p in equation (19)), which reduces constraint (9) to the following:

$$I_p + Q_p + Y_p \geq D_p \qquad \forall p \tag{19}$$

The simplifying assumption allows the problem to be solved easily using a network linear programming formulation, to be presented below, which has several advantages. First, it is mathematically efficient and solves very quickly. Second, it has intuitive appeal to managers, since the network representation closely resembles the actual process that they manage. And third, it can be expanded easily to solve large, multi-period, multi-product problems. The trade-off for these advantages, as seen in the results of Chapter 5, is in the quality of the solution. For several reasons which will be noted below, ReNet typically results in a slightly higher-cost solution, particularly for high levels of uncertainty.

The remanufacturing process depicted in Figure 3-1 begins with used cores entering the system at node 1. Cores are either disassembled (i.e. moved to node 2) or held in inventory (dotted line). Disassembled cores yield sets of used parts, some of which are usable and some of which are not. Usable parts move to node 4, while unusable parts are scrapped. A combination of usable parts (node 4) and new parts (entering into node 5) is then used to assemble units to meet demand, with any unused parts held in inventory (dotted lines). Mathematically, the process is modeled as a

generalized network linear programming model, which is easily solvable. The network model notation is defined in Table 3-2, followed by the formulation and discussion.

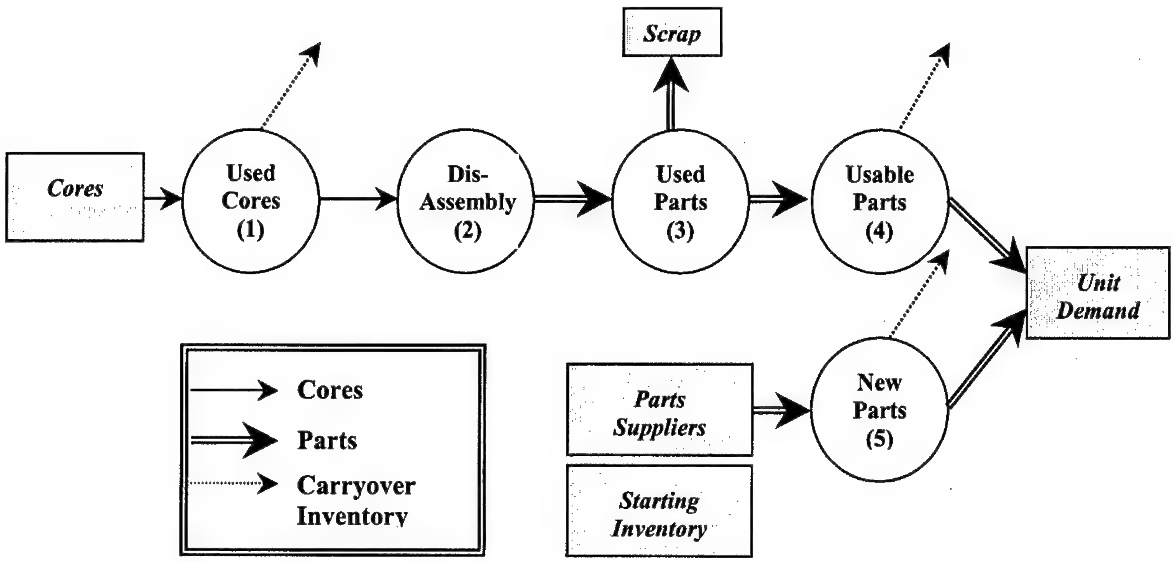


Figure 3-1: Network Representation of Remanufacturing Process

Table 3-2 Definition of Network LP (ReNet) Model Notation

X_{ijp}	=	Arc from node *I* to node *j*, part *p*
c_{ijp}	=	Cost incurred moving from node *I* to node *j*, part *p*
z_p	=	Quantity per assembly for part *p*
γ_p	=	Average yield % for part *p*
D^u	=	Demand for units
D_p	=	Demand for part *p*
S^p	=	Supply of part *p* (note: *p* is a core for node 1)
Network Arc Equivalents		
$X_{1\text{-}2}$	=	Q^d
$X_{3\text{-}4p}$	=	$\Psi_p\ (= \gamma_p Q^d)$
$X_{PS\text{-}5p}$	=	Q_p
$X_{SI\text{-}5p}$	=	I_p

$$
\begin{array}{llll}
\text{MIN} & \displaystyle\sum_i \sum_j \sum_p (c_{ijp}) X_{ijp} & & (20) \\
\text{ST:} & \text{Flow In} = \text{Flow Out} & \text{All nodes} & (21) \\
 & \left\{ X_{4Dp} + X_{5Dp} = z_p D^u \right\} \text{Demand Constraint} & \text{Demand } (k = 4 \ \& \ 5) & (22) \\
 & \left\{ X_{34p} - \gamma_p X_{23p} = 0 \right\} \text{Yield Constraint} & \text{Inspection node } (k = 3) & (23) \\
 & \left\{ \begin{array}{l} X_{12} \leq C^d \\ \dfrac{X_{4Dp} + X_{5Dp}}{z_p} \leq C^a \end{array} \right\} \begin{array}{l}\text{Capacity} \\ \text{Constraints}\end{array} & \begin{array}{l} \text{Arcs to node } (3) \\ \text{Arcs from nodes } (4 \ \& \ 5) \end{array} & \begin{array}{l} (24) \\ (25) \end{array} \\
 & \forall X \geq 0 & & (26)
\end{array}
$$

The objective function (20) minimizes the total cost, which is the sum of the costs to purchase and disassemble cores, order new parts, and hold or dispose of any excess inventory for one period. Constraints (21) are network balance constraints that apply to all nodes, and are specific to the network solution methodology. Constraints (22) ensure that demand requirements are met (a complementary supply constraint is omitted due to the assumption of infinite core and part supplies). Constraints (23) enforce the deterministic yield (and scrap) rates from the disassembly process. In combination, constraints (22) and (23) satisfy (19) by ensuring that the expected yield of each part travels to node 4 (constraint 23) and the demand for each part is met from a combination of starting inventory, used parts, and new parts (constraint 22). Constraints (24) and (25) impose the disassembly and assembly capacities, respectively, while constraints (26) limit all flows to non-negative values.

Step Two: Addition of Safety Levels

Step two begins with the deterministic solution from step one, which includes the number of disassemblies Q^d (arc $X_{1\text{-}2}$ in the network formulation) and the order quantities for each part $\{Q_p, p = 1,\ldots,P\}$ (arcs $X_{pS\text{-}5}$). Since the stochastic yields must now be accounted for (for the purposes of this research, they are modeled as rounded independent normal random variables), safety levels need to be added. Several techniques are available for calculating safety stock, the most common of which uses the standard deviation of the demand distribution. The problem with this technique is that it assumes that all of the variability in demand is due to uncertainty, whereas in most cases at least part of it is due to variations in demand that are known in advance. For this reason, several researchers have advocated using the probability distribution of the forecast error to calculate safety stock, in lieu of the distribution for demand, particularly in MRP settings (Meal 1979; Guide & Srivastava 1997; Zinn & Marmorstein 1990). Conceptually, the forecast error distribution described above translates to the probability distribution of yields for the current problem. A simple sorting technique, described below, is also added to the heuristic to take advantage of a special property of the problem and reduce the overall cost. The heuristic is now described in more detail.

Beginning with the target service level for the end-item (TSL_e), the target service level of each part TSL_p is calculated by the following.

$$TSL_p = [TSL_e]^{\frac{1}{P}} \tag{27}$$

This ensures that the product of the part service levels will meet the end-item target service level, as required by constraint (9). Since the problem is discrete in nature, however, the actual service level of each part will rarely be identical to its target service level. Instead, the last unit added will typically result in a service level somewhat higher. If the safety level for each part were calculated independently, the service level of the end item would be unnecessarily high and, by extension, more expensive. The heuristic takes advantage of this potential "surplus service level" by adding safety quantities to each part sequentially beginning with the lowest-cost item and working upward. This additional step will often reduce the quantities of higher-cost items needed and, by extension, the total cost of the solution. The specific steps of the heuristic are as follows.

1. Calculate the target service level for the parts as in equation (25)
2. Sort parts into ascending order of purchase cost
3. For each part p, beginning with the least-cost and proceeding in ascending order of cost, do the following:
   (i) Increase order quantity Q_p by one unit
   (ii) If $\Pi\, P[\psi_p \geq D_p - I_p - Q_p] \geq TSL_e$, then go to Step 4
   (iii) Else If $P[\psi_p \geq D_p - I_p - Q_p] \geq TSL_p$ and $p < P$, then go to next part
   (iv) Else If $P[\psi_p \geq D_p - I_p - Q_p] \geq TSL_p$ and $p = P$, then go to Step 4
   (v) Else go to Step 3(i)
4. End

The solution for the example using ReNet is shown below, with a total cost of \$1,858.67. The quantities in the fourth column denote the network LP solution from step one, while the buffer quantities are the additional quantities derived from the safety level heuristic described above. In this case, the addition of the simple sorting procedure saved the purchase of one unit of part 2, which ended at a service level of 98.6% instead of the

calculated 99.43% because the end-item service level met its target. Without the sorting procedure, the solution cost would have been $30 greater and the end-item service level would have been unnecessarily driven above 95%.

Table 3-3: ReNet Solution to Example Problem

Part	***Yield***	***Cost***	***Deterministic Solution***	***Buffer Qty.***	***Purchase Cost***	***Exp. Excess***	***Excess Cost***
Core	-	20.00	71	-	1420.00	-	-
1	0.9	50.00	0	0	0.00	13.9	0.40
2	0.7	30.00	0	5	150.00	4.7	0.08
3	0.6	8.00	7	6	104.00	5.6	0.03
4	0.9	4.50	0	0	0.00	13.9	0.04
5	0.5	3.50	14	6	70.00	5.5	0.01
6	0.3	3.00	29	5	102.00	5.3	0.01
7	0.9	0.50	0	0	0.00	13.9	0.00
8	0.5	0.35	14	6	7.00	5.5	0.01
9	0.3	0.15	29	5	5.10	5.3	0.00

SIPR: Stochastic Inventory Planning for Remanufacturing

Unlike ReNet, the second technique developed to solve the remanufacturing inventory problem explicitly models stochastic yields. The stochastic inventory planning model for remanufacturing (SIPR) finds a near-optimal solution using an efficient search algorithm that exploits the structure of the problem. SIPR offers two advantages over ReNet. First, while ReNet solves Q^d for the reduced deterministic problem and then holds it constant as safety quantities are added to each Q_p, SIPR solves for Q^d and the set $\{Q_p\}$ simultaneously, so that the number of disassemblies is part of the optimization search. This effectively allows additional disassemblies to be used in lieu of safety levels of new parts, where it makes economic sense to do so. Second, SIPR uses a greedy

algorithm to find the set $\{Q_p\}$ that results in a near-optimal solution for each value of Q^d in the search. So even if the same number of disassemblies is used for each, SIPR should find a better solution in most cases. There will be exceptions to the latter, however, which will be discussed in more detail later.

Like ReNet, SIPR has two steps, although in this case they are carried out simultaneously in the form of a nested loop. First, it loops through a range of values of Q^d. Next, for each value of Q^d it uses a greedy algorithm to find the optimal or near-optimal solution for the set Q_p before moving on to the next Q^d. In doing so, it stores a set of solutions and later chooses the best from the set. The general steps of the algorithm are shown below.

1. For $n = Q^d_{min}$ to Q^d_{max}, do the following
   - (A) Repeat the following until end-item Service Level ≥Target Service Level
     - (i) For each part p, do the following
       - (a) Calculate the change in service level (ΔSL) associated with a one-unit increase in Q_p
       - (b) Calculate the benefit-to-cost ratio (ΔSL/ΔCost)
     - (ii) Increase Q_p by one unit for the part p with the greatest ΔSL/ΔCost
   - (B) Save solution
   - I If n = Q^d_{max}, go to Step 2. Else increment n and go to Step 1(A)
2. Choose best solution from set $\{n\}$

There are two major design issues that must be dealt with in the SIPR algorithm. First, a range of values for Q^d must be generated that ensures the optimal value is included in the search, but does not add unnecessarily to the computation time. Second, the embedded algorithm that determines the set $\{Q_p\}$ associated with each Q^d must

generate an optimal (or near-optimal) solution and must do so efficiently, given the large number of potential solutions. These two issues are addressed in the more detailed discussion that follows.

Selection of a Search Range for Q^d

The number of values for Q^d included in the search is a key determinant of the solution time, so there is an incentive to search as narrow a range as possible. At the same time, there must be some degree of confidence that the optimal value is included in the set. This leads to a delicate balance in determining upper and lower bounds. In most realistic cases, the deterministic value can be used as a lower bound, since the end-item service level for the deterministic solution will be well below the target service level. In this case, either Q^d or some set of Q_p's, or both, will need to increase to reach the target.

Two relatively rare exceptions to this rule of thumb were noted in the experiments. First, in some extreme cases the solution crosses the "manufacturing frontier", at which point it becomes more cost-effective to simply manufacture new items rather than remanufacture used cores. In this case, the deterministic determination of Q^d (and therefore ReNet) will fail to detect the situation and will still calculate the required number of disassemblies to meet demand. SIPR, if the range for Q^d is broad enough, will account for this situation by disassembling fewer cores. The second exception is related, and is due to the level of uncertainty associated with yields. Again, in some extreme cases (illustrated in Chapters 5 and 6) higher levels of uncertainty decrease the economic value of disassemblies to a point where the optimal quantity actually decreases to a level below that of the deterministic solution, in favor of the more reliable purchase of new

parts. With these exceptions in mind, the algorithm uses the deterministic value as the lower bound, but includes checks to ensure detection of exceptions.

The selection of a maximum value for the Q^d search range is a bit more problematic. The maximum tends to be highly problem-specific and relates largely to the product characteristics and level of uncertainty, as shown in Chapter 5. In lieu of a fixed maximum, a stopping rule is used that determines a point at which the cost is continually increasing in Q^d. This heuristic makes use of the property that there is some global minimum for Q^d, beyond which the marginal value of additional disassemblies is lower than the marginal cost and the total cost is therefore monotonically increasing. The major obstacle in implementing this approach is the existence of many local minima along the curve leading to the optimum. After an extensive set of pilot runs, a stopping rule using a three-period moving average, increasing for five consecutive periods, was determined to adequately detect local minima and preclude the search's premature termination. Figure 3-2 illustrates this point using one of the problems from the experimental problem set.

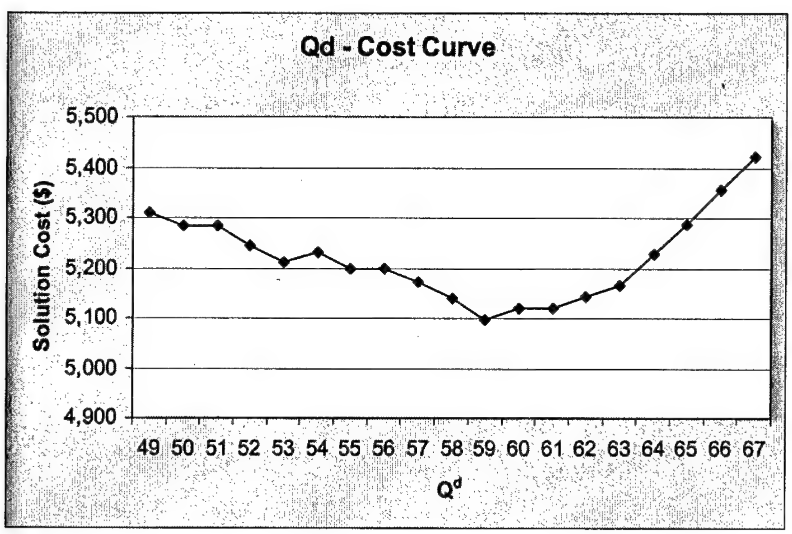


Figure 3-2: Illustrative Q^d – Cost Curve (Problem #3 from experimental problem set)

Solution of the Set $\{Q_p\}$ for Each Q^d

The solution of the optimal set of order quantities $\{Q_p\}$ is potentially much more problematic than the search range of Q^d. As previously noted, the number of potential solutions can run in the millions with just a few parts, and increases exponentially from there. SIPR therefore exploits the properties of the problem and uses a greedy algorithm to find the optimal set quickly. Although the structure of the problem is slightly different, the technique is similar to the single-site model developed by Feeney and Sherbrooke (1966). The model formulation and notation are given below, corresponding directly to the general problem formulation given by equations (8) through (13) earlier in the chapter and simplified to the single-period, single-product case.

Table 3-4: SIPR Model Notation

Decision Variables		
Q^d	=	Quantity of cores disassembled
Q_p	=	Quantity of part *p* purchased
State Variables		
D_p	=	Demand for part *p*
I_p	=	Inventory of part *p* at the start of period
D^u	=	Demand for units
ψ_p	=	Yield (in units) of part *p*
Parameter Variables		
γ_p	=	Average yield % of part *p*
c_p	=	Cost of new replacement part *p*
c_d	=	Cost of core disassembly
c_c	=	Unit cost of cores
c_{hp}	=	Cost of holding one unit of *p* for one period
C^d	=	Disassembly capacity
C^a	=	Assembly capacity
S^c	=	Available core supply (units)

$$\text{MIN} \quad (c_c + c_d)Q^d + \sum_p c_p Q_p + \sum_p (I_p + Q_p + E[\Psi_p] - D_p)^+ c_{hp} \tag{28}$$

$$\text{ST} \quad \prod_p \Pr[(I_p + \Psi_p + Q_p) \geq D_p] \geq TSL \tag{29}$$

$$Q^d \leq C^d \tag{30}$$

$$Q^d \leq S^c \tag{31}$$

$$D^u \leq C^a \tag{32}$$

$$Q^d, Q_p \qquad \text{Positive integers} \tag{33}$$

The objective function (28) minimizes the sum of core purchase and disassembly, new part purchase, and holding (or disposal) costs, as before. Yield uncertainty is explicitly modeled via constraint (29), and is discussed in more detail below. Constraints (30) and (32) enforce the disassembly and assembly capacities, respectively, while constraints (31) enforce the core supply availability. Constraints (33) ensure all decision variables are positive integers.

Service Levels of Individual Parts. To demonstrate the mechanics of the greedy algorithm, it is first necessary to examine the properties of the service level function for individual parts. To clarify the discussion, constraint (34), which is a more detailed version of constraint (29) above, is given below.

$$\prod_p \Pr[(I_p + Q_p + \Psi_p) \geq D_p] \geq TSL_p \tag{34}$$

The left-hand side of constraint (34) is the end-item service level, defined as the probability of having enough complete sets of parts on hand to meet end-item demand, and is computed as the product of the service levels of all constituent parts. The service level of each part is further defined as the probability of having an adequate quantity on hand, comprised of starting inventory (I_p), new parts (Q_p), and used parts from cores (Ψ_p, a random variable), to meet individual demand. Rearranging the terms, the service level of part p (SL_p) can be rewritten as in (35) below.

$$SL_p = \Pr[\Psi_p \geq (D_p - I_p - Q_p)] \tag{35}$$

Since the problem is discrete, (27) is more specifically written as equation (36).

$$SL_p = \sum_{x=D_p-I_p-Q_p}^{Q^d} \Pr[\Psi_p = x] \tag{36}$$

Since D_p, I_p, and Q^d are all specified at this point in the algorithm, the sole determinant of SL_p is Q_p, the decision variable of interest. Any increase in Q_p will broaden the limits of the summation and therefore lead to a non-negative change in the service level. In fact, the change will always be positive unless the service level is 1. The service level is therefore monotonically increasing in Q_p, and by extension in cost, since the cost is very nearly linear in Q_p (there is a small holding cost component that is non-linear in Q_p, but its magnitude relative to the total is negligible in any practical

application). The service level is illustrated in Figure 3-3 below using a binomial distribution for the yield Ψ_p.

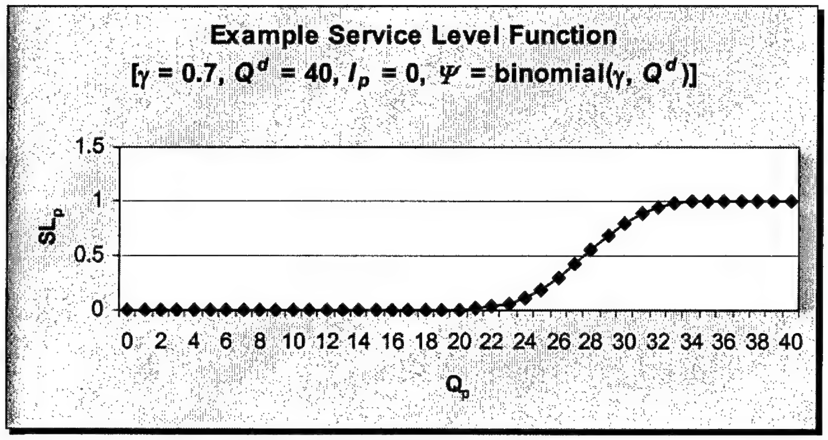


Figure 3-3: Illustrative Service Level Function for a part p

For the purpose of an optimization search, a concave function is mathematically more convenient than that of Figure 3-3. For this reason, and another that will soon become apparent, the logarithmic transformation of the service level is used by SIPR. Given the form of the service level function, its logarithm is concave in $0 \leq SL_p \leq 1$, as shown in Figure 3-4.

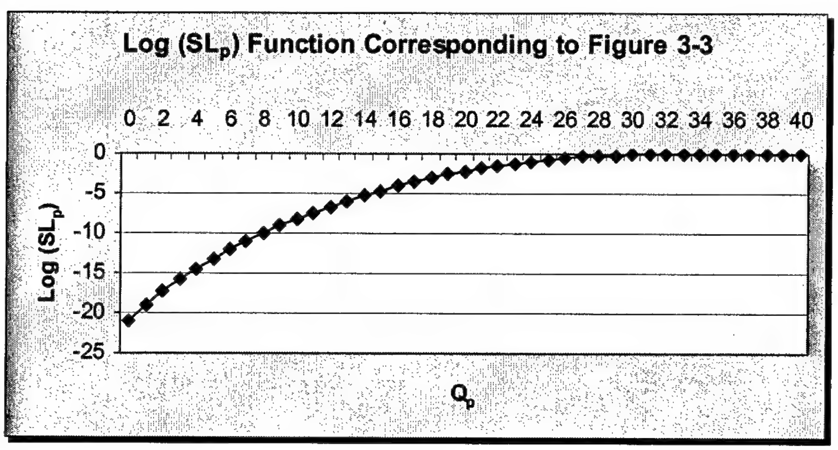


Figure 3-4: Logarithm of Service Level function from Figure 3-3

End-Item Service Level. The end-item service level, as defined by (34), is equal to the product of the service levels of its constituent parts. To exploit the concavity of the logarithmic transformation of the part service levels, however, the end-item service level must be an *additive separable* function of the part service levels. Fortunately, the logarithmic transformation, in addition to being a concave function at the part level, also conveniently allows the transformation of the end-item service level function to a similar form that is both concave and additive separable.

$$Log(SL) = Log(\prod_p SL_p) = \sum_p Log(SL_p) \tag{37}$$

Since the logarithms of the part service levels are concave, their sum (and therefore the logarithm of the end-item service level) is also concave and additive separable. This result allows the use of a greedy, or "steepest ascent" algorithm to find the optimal solution, since a function and its logarithm achieve their maxima at the same point. For each value of Q^d, then, the greedy algorithm is used to traverse the efficient hull of the logarithm of the service level function to efficiently find the optimal set $\{Q_p\}$. Specifically, at each step the slope is calculated for each part p by (38).

$$\left[\frac{\Delta Log[SL]}{\Delta Cost}\right]_p = \frac{Log[SL(Q_p+1)] - Log[SL(Q_p)]}{Cost(Q_p+1) - Cost(Q_p)} \tag{38}$$

Potential for Non-Optimal Solutions. As alluded to earlier, the algorithm described in the previous section can not guarantee an optimal solution, despite the convenient form of

the logarithm of the service level function. The reason relates not to the algorithm itself, but to the discrete nature of the problem. At each step, the algorithm chooses to increase the order quantity by one of that part offering the greatest increase in service level divided by its associated cost, i.e. the one with the steepest slope. This ensures that only the most efficient solutions are selected at each step. When the target service level is near, however, there is a possibility of "inefficient" solutions that represent a lower-cost step, but still reach the target service level. Fortunately, the log function is very flat in this region and therefore the area in which one of these alternate solutions can lie is small, as shown in Figure 3-5. So even a non-optimal solution generated by SIPR is likely to be very close to optimal. A pilot study reported in Chapter 5 compares SIPR solutions with those of an implicit enumeration algorithm to estimate the frequency of occurrence of sub-optimal solutions, as well as their relative magnitude.

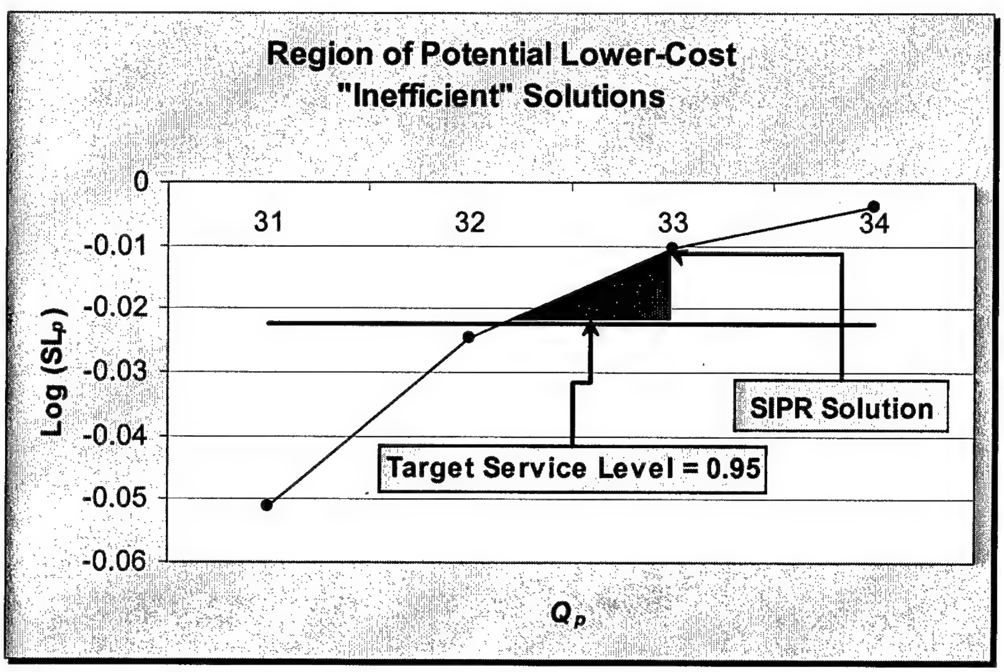


Figure 3-5: Illustration of the region in which potential lower-cost solutions can lie

Applying the SIPR algorithm to the example from earlier in the chapter, the optimal solution is attained (Table 3-5). The total cost of the SIPR solution, using a target service level (TSL) of 0.95 as before, is $1799.82. Recall that the ReNet solution resulted in a total cost of $1858.67, about 3% higher than that of SIPR. The performance gap for this example problem is consistent with the results of the main experiment discussed in Chapter 5. The next chapter discusses the details of the experimental design.

Table 3-5: SIPR Solution to Example Problem

Part	***Yield***	***Cost***	***Order Qty.***	***Purchase Cost***	***Exp. Excess***	***Excess Cost***
Core		20.00	78	560.00		
1	0.9	50.00	0	0.00	20.2	0.58
2	0.7	30.00	0	0.00	4.6	0.08
3	0.6	8.00	9	72.00	5.8	0.03
4	0.9	4.50	0	0.00	20.2	0.05
5	0.5	3.50	17	59.50	6.0	0.01
6	0.3	3.00	32	96.00	5.4	0.01
7	0.9	0.50	0	0.00	20.2	0.01
8	0.5	0.35	18	6.30	7.0	0.00
9	0.3	0.15	35	5.25	8.4	0.00

CHAPTER 4

EXPERIMENTAL DESIGN

This chapter presents the major research questions and describes the numerical experiment designed to answer those questions. Included in the discussion are details on the performance measures, experimental factors and levels, and experimental design.

Research Questions

The main experiment described in this chapter is designed to answer the following three primary research questions.

1. ***To what extent, and under what conditions, does a planning approach that accounts for stochastic yields outperform a deterministic (network LP) approach with respect to cost?*** Given that stochastic techniques are computationally expensive relative to their deterministic counterparts, it is important to know how each performs in a variety of situations. In answering this question, the conditions under which a more sophisticated technique is warranted are better understood. Although the main experiment is limited to the single-period case, a limited secondary experiment tests both approaches in solving the multi-period problem.

2. ***What effect does the reduction of yield uncertainty (i.e. prior knowledge of expected yields) have on cost?*** As previously discussed, the most significant (and unique) challenge faced by inventory planners in remanufacturing is the uncertainty of yields. Yields depend, at least in part, on the condition of the cores being disassembled. So prior knowledge about the condition of cores should allow inventory planners to forecast part yields with greater accuracy and, therefore, reduce uncertainty and its associated cost. The experiments attempt to quantify the value of different levels of prior knowledge.

3. ***How do a product's structural characteristics (yield percentages, part costs, and number of parts, e.g.) affect the cost of the solution?*** The structure of a product has a profound impact on the inventory cost of remanufacturing. Specifically, four primary factors drive the solution: the range of yield percentages, part costs, the relative match between the yields and costs, and the total number of parts in the end-item. Yield percentages are significant because a wide range of expected yields increases the difficulty of attaining a low-cost solution. For example, at one extreme, if all parts have the same yield the difficulty of determining the optimal Q^d is essentially eliminated and order quantities are straightforward. Likewise, a wide range of part costs makes the solution less intuitive. In addition to the yields and costs themselves, the "matching" of the two can play a significant role. Clearly, a case where the highest-cost part also has a high yield is considerably less costly from an inventory standpoint than a case where the same high-cost part has a low yield. And finally, since the overall service level

involves the product of all individual part service levels, more complex products with a higher number of parts would seem be more costly in terms of inventory. These four factors were used to generate a set of problems that encompasses a wide range of product structures. The problems were then solved and compared in the numerical experiment.

Performance Measures

Solution Cost

The primary performance measure of interest is the total cost of a solution. The total cost includes the costs of purchasing cores, disassembling cores, purchasing parts, and holding or disposing of inventory. The cost of lost sales is not explicitly included, but it is implicitly assumed that the target service level, although fixed in this experiment, can be fixed at a level that will minimize or avoid lost sales. Assembly and distribution costs are omitted, since they are the same regardless of the inventory planning technique employed. In one of the sensitivity experiments, the total cost is divided by demand to calculate a per-unit cost, since the demand is varied in this case and direct comparisons using total cost are not possible.

Performance Gap

In answering the first research question, which compares the ReNet and SIPR solutions, the performance measure of interest is not the total cost of the solutions but the cost gap between the two techniques. Although directly derived from the solution costs, the measure is reported as the Performance Gap, which represents the percent difference

between the optimal/near-optimal solution (SIPR) and the heuristic approximation (ReNet). This measure is further broken down into two components representing the two theoretical advantages of SIPR: the advantage of the greedy algorithm in finding an optimal solution and the advantage of the ability to disassemble more cores in lieu of buying safety stock.

Solution Time

In addition to the cost-based measures, the solution times are reported as a proxy for problem complexity. As previously discussed, stochastic models are computationally more difficult than their deterministic counterparts, so the extent to which their complexity translates to solution time is relevant in the context of the comparison. Little emphasis is placed on the actual magnitude of solution times, since in practice it is highly dependent on biasing factors like computer equipment and programming sophistication. All runs were completed on a 1.7 MHz microprocessor with 256 MB random access memory.

Optimal Number of Disassemblies

A final measure used in certain cases, particularly for the sensitivity experiments, is the optimal number of disassemblies Q^{d*}. Although it is a decision factor in both models, its use as a secondary performance measure often adds insights that are not possible through the use of solution cost alone. The results of the sensitivity experiments also shed light on the effects of limited core supply and/or disassembly capacity, either of which reduces the cost savings associated with SIPR.

Experimental Factors

The experimental factors were chosen with two considerations in mind. First, they are designed to answer the research questions. And second, they are limited to a reasonably-sized set to make the experiment both feasible and meaningful. With that in mind, the following experimental factors were used.

Inventory technique (2 levels)

Regardless of the level of uncertainty, either ReNet or SIPR can be employed. This factor includes two levels corresponding to the two approaches described in Chapter 3, and is designed to answer research question 1. In order to make a legitimate comparison, ReNet will include safety stock levels based on the yield forecast error, similar in concept to that of Meal (1979), which was modified and tested by Guide and Srivastava (1997) for a remanufacturing environment. It is added using a simple heuristic that can improve its cost performance, as described in Chapter 3.

Yield Uncertainty (3 levels)

To answer research question 2 regarding the value of prior knowledge of yields, a yield uncertainty factor is included in the experiment. The theoretical basis for this factor relates to the assumption that some of the variance experienced in yields relates to the condition of the specific cores being disassembled. As such, yields can be more accurately forecasted with increased knowledge. More specifically, the varying levels of uncertainty can be thought to correspond to varying levels of prior knowledge about core condition, as illustrated below:

Table 4-1: Levels of Uncertainty

Level of Uncertainty	***Level of Prior Knowledge about core condition***
High	None: No prior knowledge of core condition
Moderate	Low: Estimated condition (based on age, visual inspection, or usage history, e.g.)
Low	Moderate: Diagnosed condition (diagnostic testing, e.g.), but random sampling error remains

Note that the levels of uncertainty are suggested by operational uncertainty-reduction techniques of different types. For example, a simple measure like the age of a product can be used to reduce the uncertainty of yields, since older products will tend to yield fewer usable parts. Better still is the use of diagnostic test equipment like those used on automobiles and military weapon systems, which can often pinpoint the exact components that are operating below tolerance without disassembling the unit. An extreme case of no uncertainty could be added to correspond to a policy of disassembling all cores and testing all individual parts prior to ordering new parts. This case is omitted from the experiment, however, since it is a trivial case for which ReNet will always find the optimal solution. By testing a wide range of possible values at three levels, better insights can be drawn regarding the value of prior information.

The level of uncertainty is operationalized in the experiment as the standard deviation of the yield distributions. In the main experiment, it is varied at three levels corresponding to standard deviations of 0.01, 0.03, and 0.05. Note that these are applied to the average yield percentages γ_p, as opposed to the yield quantities Ψ_p. The levels were selected in order to cover the widest range possible, with the constraint that the tails of the normal distribution needed to be included in the feasible region for the total yields. This constraint precludes any infeasible quantities from being included in the analysis, as

would be the case for a part with an average yield of 0.85 and a standard deviation greater than 0.05.

Product Structure (4 factors, 2 levels each)

Using the four characteristics of product structure previously described, a set of 16 problems was developed and solved to determine their effects on the cost of the solution (Table 4-2). In general, the levels were selected to represent the two extremes in each case, so that the entire range of reasonable values is included. For example, the cost profile factor is bounded by a balanced profile, in which all parts have roughly equivalent costs and a Pareto or "ABC" profile, in which a small percentage of the parts accounts for the majority of the total cost. Two levels were used for the number of parts factor in order to keep the problem small and tractable while maintaining the ability to test the effect of doubling the number of parts. Likewise, the levels of the yield range factor were limited somewhat by the necessity of capturing all of the probability associated with stochastic yields, as described in the previous section. Specific details of the product structure factors are discussed below.

Table 4-2: Parameters of Experimental Problem Set

<table>
<tr><th>PRODUCT STRUCTURE CHARACTERISTIC</th><th>TREATMENT LEVEL</th><th>DESCRIPTION AND VALUES</th></tr>
<tr><td rowspan="2">Yield Range</td><td>Narrow</td><td>Average yields distributed evenly between 0.55 and 0.85</td></tr>
<tr><td>Wide</td><td>Average yields distributed evenly between 0.15 and 0.85</td></tr>
<tr><td rowspan="2">Part Cost Profile</td><td>Balanced</td><td>Distributed evenly</td></tr>
<tr><td>ABC</td><td>Various distributions following Pareto (ABC) profile<table><tr><th>% of Parts</th><th>% of Total Cost</th></tr><tr><td>20</td><td>80</td></tr><tr><td>40</td><td>18</td></tr><tr><td>40</td><td>2</td></tr></table></td></tr>
<tr><td rowspan="2">Yield-Cost Match</td><td>High-High</td><td>High cost parts have highest yields</td></tr>
<tr><td>High-Low</td><td>High cost parts have lowest yields</td></tr>
<tr><td rowspan="2">Number of Parts</td><td>High</td><td>10</td></tr>
<tr><td>Low</td><td>5</td></tr>
</table>

Two ranges of yields are included, representing a narrow range from either 0.15 to 0.45 or 0.55 to 0.85 (Narrow), and a wider range from 0.15 to 0.85 (Wide). The range of yields should not be confused with the standard deviation of the yield distribution. The latter is a parameter of the uncertainty of the actual yields of each part coming from cores, given their averages. The former describes the range of those average yields across the different parts in a product. The bounds of the wide range, as previously discussed, were selected so that the normal probability distribution of the part yields would be contained within the feasible region. In other words, for a part with an expected yield of 0.85 and a standard deviation of 0.05, the probabilities are contained (almost) entirely within the range of 0.6 and 1.0. A higher expected yield would result in part of the probability distribution extending into the infeasible range beyond 1.0.

Part costs are distributed as either balanced or skewed in a manner consistent with the typical Pareto or "ABC" analysis of parts (ABC). The latter is far more common in practice, particularly for more complex products like engines, transmissions, and heavy machinery, while the former is sometimes found in simpler products in which there are several relatively inexpensive parts, like single-use cameras.

The parts are assigned their associated yields and costs in two ways. The first represents the case where high-cost parts also have the highest yields and vice-versa (High-High) and the second where high-cost parts have the lowest yields (High-Low). Although the latter is included for experimental completeness, it is rarely if ever found in practice. As the results in Chapter 5 will demonstrate, this case is generally not economically viable, as should be expected.

Finally, two products are tested using totals of 5 and 10 parts, respectively. The lower value of 5 parts was chosen as the simplest product for which an ABC cost profile is possible. In this case, 20% of the parts (i.e. 1 part) account for 80% of the cost. The higher value of 10 parts maintains a tractable problem size, but doubles the number of parts, making meaningful comparisons possible. For all problems, the expected value of parts emanating from the first disassembled core is held constant to ensure unbiased comparisons in the numerical experiment.

Fixed Factors

Several factors were fixed in the experiment in order to avoid confounding effects. While a combination of real-world examples and the results of pilot runs were used to set the levels at reasonable values, the sensitivity experiments further examine some of their effects individually. First, the target service level was fixed at 0.95 for both the single- and multi-period experiments, although its effect was tested in a sensitivity experiment. Second, the cost of purchasing and disassembling a core was set at $80, a value roughly equivalent in relation to the part costs to that of an actual transmission case studied during the course of this research. Once again, its effect on the performance measures was tested in a sensitivity experiment. Third, end-item demand was held fixed at 50 units for the main experiment and was varied in a sensitivity analysis. And finally, the problems were run as unconstrained with respect to assembly and disassembly capacities and core supply. The unconstrained problem, while somewhat unrealistic, allowed the properties of the problem to be fully explored. In any case where one of these constraints is binding, the cost of the SIPR solution would increase because the optimal value of Q^d would lie in the infeasible region.

Experimental Design and Analysis

The experiment is designed as a numerical experiment, with a focus on answering the research questions in the context of the experimental factors. Since core supply and end-item demands are held constant, a more complex simulation experiment would do little to offer additional insight into the nature of the solution. In its place, an extensive set of problems is used to test the solutions across many different product types. This being the case, analysis and comparisons of the solutions are done much less formally than a simulation study would warrant. Emphasis is placed on the practical significance of differences, vice the more common statistical significance, since the latter is inappropriate for a numerical study of this type.

Main Experiment

The product characteristic factors described in the previous section were used to generate a set of 16 problems, each of which was solved using both techniques under three levels of yield uncertainty for a total of $16 \times 3 \times 2 = 96$ factor-level combinations. Figure 4-1 illustrates the experimental design, keeping in mind that each problem of the set represents one combination of the four product characteristic factors. The main experiment focuses on the three research questions, and the results presented in Chapter 5 are organized in that way. For each, the main and second-order interaction effects of each factor are investigated in the context of the questions.

Problem	Technique				SIPR			ReNet		
	Yield Uncertainty									
	# Parts	*Yield Range*	*Cost Profile*	*Cost-Yield Match*	*Low*	*Med.*	*High*	*Low*	*Med.*	*High*
N-B-HH-10	10	Narrow	Balanced	HiHi						
N-B-HL-10	10	Narrow	Balanced	HiLo						
N-A-HH-10	10	Narrow	ABC	HiHi						
N-A-HL-10	10	Narrow	ABC	HiLo						
W-B-HH-10	10	Wide	Balanced	HiHi						
W-B-HL-10	10	Wide	Balanced	HiLo						
W-A-HH-10	10	Wide	ABC	HiHi						
W-A-HL-10	10	Wide	ABC	HiLo						
N-B-HH-5	5	Narrow	Balanced	HiHi						
N-B-HL-5	5	Narrow	Balanced	HiLo						
N-A-HH-5	5	Narrow	ABC	HiHi						
N-A-HL-5	5	Narrow	ABC	HiLo						
W-B-HH-5	5	Wide	Balanced	HiHi						
W-B-HL-5	5	Wide	Balanced	HiLo						
W-A-HH-5	5	Wide	ABC	HiHi						
W-A-HL-5	5	Wide	ABC	HiLo						

Figure 4-1: Experimental Design

Sensitivity Analysis

A set of four sensitivity analyses were conducted to test (1) several factors that were held constant in the main experiment, but have an effect on the solution; and (2) the effects of levels of uncertainty outside the range included in the main experiment. The additional factors tested were end-item demand, core/disassembly costs, and target service level, which were held constant in the main experiment at 50 units, \$80, and 0.95, respectively. The level of uncertainty was varied from 0.001 to 0.11 in a sensitivity experiment. A subset of four problems was used in the sensitivity analysis for the fixed factors, while two were used for the level of uncertainty since the problems with a wide yield range limit the amount by which the standard deviation can vary.

Secondary Experiment

In addition to the main and sensitivity experiments, an abridged set of 4 multi-period problems was solved using both techniques. The main purpose of the secondary experiment was to test a heuristic version of the stochastic model vis a vis the network LP solution, the latter of which was expected to be better suited for this class of large problems. The four problems selected focus on the yield range and cost profile factors, since the number of parts showed little effect in the main experiment and the cost-yield match factor interacted so strongly with the other factors that it often biased the results. The yield uncertainty factor was held constant at the medium level, corresponding to a standard deviation of 0.03. In addition to the usual factors, four different demand patterns were used to test the effects of varying demand over time.

CHAPTER 5

PRIMARY EXPERIMENTAL RESULTS

In this chapter, the results of the primary analytical experiment are presented and discussed. The discussion begins with two pilot studies that were conducted prior to execution of the primary experiment. The first tests two local search heuristics developed to capture the optimal solution within the SIPR framework. The second compares the SIPR results to the optimal solutions derived through a partial implicit enumeration algorithm, to validate SIPR's near-optimality. Since the SIPR results are used as a baseline for the remaining analysis, it is important to have a degree of confidence that they are optimal or near-optimal, particularly when comparing ReNet's results with those of SIPR.

The results of the primary experiment, in which a set of single-period problems are solved, are discussed next. The discussion is organized according to the three research questions identified in the preceding chapter. For each, the main and interaction effects are analyzed in the context of answering the research questions, followed by a brief discussion of the implications of the results. The chapter concludes with a summary of the results and managerial implications of the primary experiment.

Results of Pilot Study #1: Local Search Heuristics

As discussed in Chapter 3 (p.42-43), SIPR does not guarantee an optimal solution even though its structure would suggest that it does. It does provide a very efficient search procedure that takes advantage of the problem structure and arrives at a solution very close to optimal. As a result, two enhancements to SIPR were developed in an attempt to capture the optimal solution in those cases where it is initially missed. Since SIPR will get a solution very close to optimal, each procedure uses the SIPR solution as a starting point. A key consideration in developing the local search heuristics was reasonable computation time. Two such procedures were developed and tested on the original set of 24 problems at three levels of uncertainty, yielding 72 total problems (note: 8 of the original 24 problems were eliminated in the main experiment, since their results offered no additional insight, leaving a set of 16 problems at three levels of uncertainty for the main experiment). The results were then compared to those of SIPR. The two procedures are described below, followed by the results of the pilot study.

Local Search Heuristic #1 – Cheapest Final Step (CFS)

The first, and simpler, of the two local search procedures involves a simple check at each step of SIPR to determine if increasing any order quantity Q_p (other than the one chosen by SIPR) yields a solution in which the service level meets or exceeds the target service level. All such feasible solutions are saved during the initial search, and their overall costs are compared with the final SIPR solution. For example, in Figure 5-1 the SIPR heuristic travels up the service level-cost curve along the efficient hull. After the third step, as before, the next step chosen is the one with the greatest benefit-to-cost ratio

(i.e. the steepest slope). In this case however, of the remaining three "inefficient" solutions, one meets the target service level at a slightly lower cost than the efficient solution because of the discrete nature of the problem. This procedure typically adds only a few feasible solutions to the search, and the additional computing time is negligible. It did improve the solutions of many problems, however, and captured several optimal solutions that SIPR had missed. It is limited in the sense that there is no guarantee that the optimal solution can be reached from one of the points on the SIPR path.

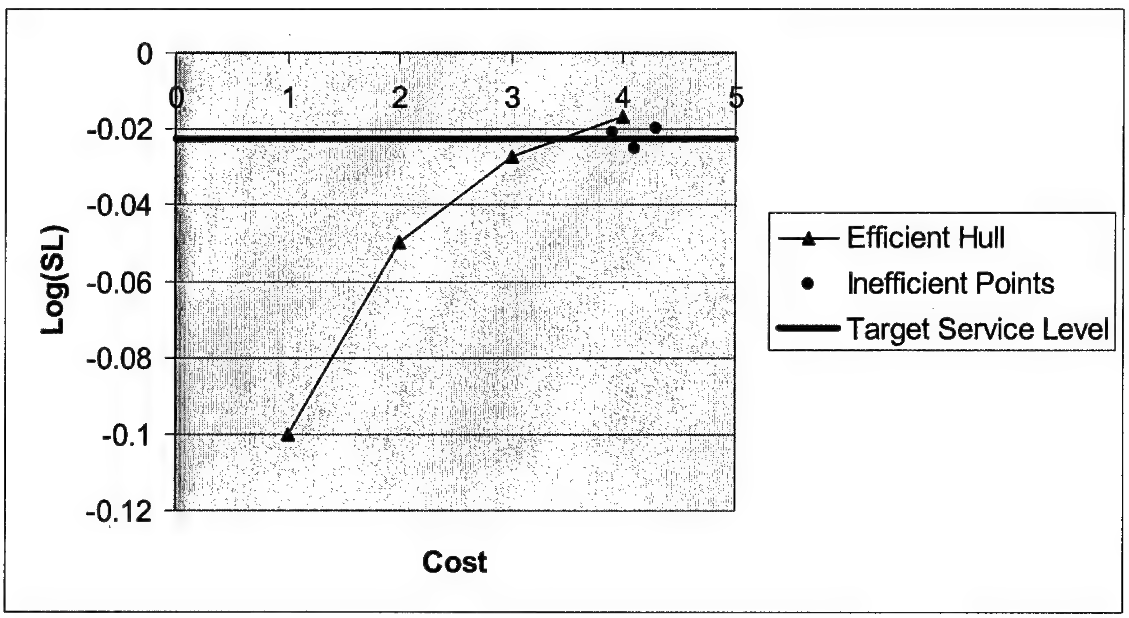


Figure 5-1: Non-optimality Example

Local Search Heuristic #2 – Partial Enumeration (PE)

The second procedure uses a more exhaustive search in which every possible solution between Q_p -1 and Q_p +1 for all p is checked and compared to the SIPR solution. Although much more rigorous, this procedure requires that an additional 3^P potential solutions be investigated, since each part is varied from -1 to +1 of the SIPR $\{Q_p\}$ solution set. For 10-part problems, this means over 59,000 additional calculations, significantly adding to the computation time. Table 5-1 shows a comparison of the solutions attained by the two additional procedures.

Table 5-1: Results of Local Search Procedures

Number of Parts	Cases Where:	Number of Cases	Average Gap	Maximum	Minimum	Average Increase in Solution Time*
10	CFS Cost = PE Cost	26	N/A			555.6%
	CFS Cost > PE Cost	10	0.025%	0.078%	0.006%	
5	CFS Cost = PE Cost	30	N/A			74.1%
	CFS Cost > PE Cost	6	0.049%	0.177%	0.014%	
Total		72	0.037%	0.177%	0.006%	314.9%

*Problems were run on a 1.7 MHz processor with 256 MB of random access memory (RAM)

From the perspective of practical significance, the meager cost improvements offered by the PE heuristic do not justify the disproportionate increases in computing time. In 56 of 72 cases, the two heuristics arrived at exactly the same solution. For the remaining 16, all differences were less than two-tenths of one percent. Conversely, the average increase in computing times associated with the PE heuristic was over 550% for the 10-part problems and 74% for the 5-part problems. For the remainder of the chapter,

all results for SIPR imply the basic SIPR heuristic with the CFS local-search heuristic added.

Pilot Study 2: Check For SIPR Optimality

Due to the discrete nature of the problem, it has been shown that SIPR will not always arrive at the optimal solution. However, it should always arrive at a solution very close to optimal, particularly with the addition of the CFS heuristic described in the previous section. To test this hypothesis, a second pilot study was conducted to compare the SIPR solutions to the optimal solutions. The computation time required to determine these optimal solutions is evidence enough of the need for an alternative approach. The 10-part problems require an average of 3.9 hours of CPU time each on a 1.7 GHz processor. In fact, if every possible combination of order quantities were to be checked, the problems would likely take weeks instead of hours, since the 10-part problem can have trillions of possible combinations.

The procedure used here begins the search with each part's order quantity set at the minimum level that gives it a service level at least equal to the end-item target service level. The reason for the starting point is simply that if any individual part has a service level less than the target end-item service level, the target service level cannot possibly be met. Even with this initialization, millions of combinations must be explored to ensure an optimal solution. The cases in the pilot study represent the optimal solutions of each problem for each value of Q^d. As such, a number of different cases are included for each problem. A total sample of 384 cases were tested, distributed across all problems and the three levels of uncertainty. Table 5-2 shows the results of the comparison.

Table 5-2: Results of Optimality Check

Result	# Cases	% Cases	Gap ($)			Gap(%)		
			Avg.	*Max.*	*Min.*	*Avg.*	*Max.*	*Min.*
Optimal	370	96%	0.00	0.00	0.00	0.000%	0.000%	0.000%
SubOptimal	14	4%	8.87	48.00	1.00	0.011%	0.210%	0.010%
Total	384	100%	0.30	48.00	0.00	0.002%	0.210%	0.000%

As with the PE heuristic described in the previous section, the additional computation time required to find the optimal solution does not appear to be justified. Only in about 4% of the cases tested did the implicit enumeration algorithm find a better solution than SIPR, and the average gap of those problems was less than 1/100th of one percent. The maximum gap was only about 0.2%. Based on the problem set used in this experiment, it appears that the SIPR solution is optimal in most cases, and near-optimal in the few that remain.

The remainder of the chapter focuses on the results of the primary experiment. All factor-level combinations described in Chapter 4 are included: a total of 16 problems (4 product structure characteristics at 2 levels each), run at 3 levels of uncertainty, by each of the two techniques, for a total of 96 runs. As discussed in Chapter 4, the levels chosen for each factor attempt to capture the two opposite extremes of each, so that inferences can be made regarding the characteristics and behavior of the remanufacturing inventory planning problem. In lieu of formal statistical testing, informal pair-wise comparisons are made between like problems. The results of a set of associated sensitivity analyses and the secondary (multi-period) experiment are discussed in Chapter 6.

Research Question 1

To what extent, and under what conditions, does a planning approach that accounts for stochastic yields outperform a deterministic (network LP) approach with respect to cost?

Summary Performance Comparison

The results of the main experiment are shown in Table 5-3 below, followed by a discussion of the overall results. Detailed analysis follows in the next section.

Table 5-3: Summary of Results for SIPR and ReNet

Number of Parts	Problem	Level of Uncertainty	SIPR			ReNet			Corr. SIPR Cost (same Q^d)	Cost Gap (% ReNet over SIPR)			Time Gap (s)
			Cost	Q^d	Time (s)	Cost	Q^d	Time (s)		Due to Heuristic	Due to Q^d Change	Total	
10	N-B-HH-10	Low	5,935	63	39.4	6,014	59	0.6	5,982	0.54%	0.78%	1.33%	39
		Medium	6,326	66	101.9	6,430	59	0.9	6,384	0.73%	0.92%	1.66%	101
		High	6,773	69	192.7	6,828	59	1.2	6,799	0.43%	0.38%	0.82%	192
	N-B-HL-10	Low	15,899	126	298.4	16,021	119	0.6	15,916	0.66%	0.11%	0.77%	298
		Medium	17,200	0	9.1	18,081	119	1.2	17,985	0.53%	4.56%	5.12%	8
		High	17,200	0	10.3	20,147	119	1.7	20,084	0.31%	16.77%	17.13%	9
	N-A-HH-10	Low	5,097	59	42.3	5,146	59	0.5	5,097	0.97%	0.00%	0.96%	42
		Medium	5,381	62	116.8	5,514	59	0.8	5,434	1.47%	0.99%	2.47%	116
		High	5,682	67	211.2	5,825	59	1.0	5,748	1.34%	1.16%	2.52%	210
	N-A-HL-10	Low	25,692	176	581.7	25,990	208	0.5	25,832	0.61%	0.55%	1.16%	581
		Medium	26,982	0	9.3	31,013	208	0.9	30,411	1.98%	12.71%	14.94%	8
		High	26,982	0	9.7	36,358	208	1.4	35,348	2.86%	31.01%	34.75%	8
	W-B-HH-10	Low	8,433	61	45.0	8,492	59	0.5	8,459	0.39%	0.30%	0.70%	45
		Medium	8,968	65	115.6	9,052	59	0.8	9,009	0.48%	0.46%	0.94%	115
		High	9,533	50	64.9	9,603	59	1.0	9,568	0.36%	0.37%	0.74%	64
	W-B-HL-10	Low	9,823	68	123.3	9,877	65	0.5	9,831	0.47%	0.09%	0.55%	123
		Medium	10,467	65	234.5	10,509	65	0.7	10,467	0.40%	0.00%	0.40%	234
		High	10,681	0	10.1	11,207	65	1.0	11,176	0.28%	4.63%	4.93%	9
	W-A-HH-10	Low	5,522	59	45.2	5,569	59	0.5	5,522	0.85%	0.00%	0.85%	45
		Medium	5,819	63	124.8	5,961	59	0.8	5,869	1.57%	0.85%	2.43%	124
		High	6,150	67	225.7	6,295	59	1.0	6,210	1.37%	0.97%	2.35%	225
	W-A-HL-10	Low	19,875	100	196.3	20,033	106	0.5	20,033	0.00%	0.80%	0.80%	196
		Medium	20,810	0	434.6	22,025	106	0.7	21,754	1.24%	4.54%	5.84%	434
		High	20,810	0	10.3	24,064	106	0.9	23,731	1.40%	14.04%	15.64%	9
Averages - 10 Parts			**12,585**	**53.6**	**135.5**	**13,586**	**91.8**	**0.8**	**13,444**	**0.89%**	**4.04%**	**4.99%**	**135**
5	N-B-HH-5	Low	6,072	61	12.3	6,110	59	0.4	6,083	0.44%	0.19%	0.63%	12
		Medium	6,417	65	26.7	6,507	59	0.6	6,475	0.49%	0.90%	1.39%	26
		High	6,817	69	47.7	6,871	59	0.7	6,848	0.34%	0.45%	0.79%	47
	N-B-HL-5	Low	15,534	119	77.3	15,695	111	0.5	15,629	0.42%	0.61%	1.04%	77
		Medium	16,934	0	4.4	17,383	111	0.8	17,388	-0.03%	2.68%	2.65%	4
		High	16,934	0	3.0	19,150	111	1.0	19,084	0.34%	12.70%	13.08%	2
	N-A-HH-5	Low	5,014	60	11.3	5,042	59	0.4	5,031	0.22%	0.33%	0.56%	11
		Medium	5,271	63	29.6	5,396	59	0.6	5,301	1.79%	0.56%	2.36%	29
		High	5,539	66	52.3	5,662	59	0.7	5,582	1.43%	0.78%	2.22%	52
	N-A-HL-5	Low	27,619	218	172.8	27,839	222	0.5	27,889	-0.18%	0.98%	0.80%	172
		Medium	28,804	0	2.8	32,503	222	0.6	32,283	0.68%	12.08%	12.84%	2
		High	28,804	0	3.1	38,490	222	0.8	36,763	4.70%	27.63%	33.63%	2
	W-B-HH-5	Low	8,621	61	12.1	8,659	59	0.5	8,659	0.00%	0.44%	0.44%	12
		Medium	9,114	64	27.8	9,196	59	0.6	9,156	0.44%	0.46%	0.90%	27
		High	9,649	55	46.6	9,691	59	0.7	9,656	0.36%	0.07%	0.43%	46
	W-B-HL-5	Low	9,830	63	32.0	9,926	59	0.5	9,878	0.49%	0.48%	0.98%	32
		Medium	10,396	62	58.6	10,436	59	0.6	10,433	0.03%	0.36%	0.38%	58
		High	10,700	0	4.1	11,029	59	0.7	10,998	0.28%	2.79%	3.07%	3
	W-A-HH-5	Low	5,275	60	11.8	5,296	59	0.4	5,296	0.01%	0.39%	0.40%	11
		Medium	5,534	63	30.2	5,674	59	0.6	5,576	1.76%	0.76%	2.53%	30
		High	5,831	66	54.2	5,951	59	0.7	5,867	1.44%	0.62%	2.07%	53
	W-A-HL-5	Low	23,121	99	63.5	23,201	100	0.4	23,201	0.00%	0.35%	0.35%	63
		Medium	23,940	0	2.9	25,073	100	0.5	25,160	-0.34%	5.10%	4.73%	2
		High	23,940	0	3.2	26,995	100	0.6	26,739	0.96%	11.69%	12.76%	3
Averages - 5 Parts			**13,155**	**54.8**	**32.9**	**14,074**	**91.0**	**0.6**	**13,957**	**0.67%**	**3.47%**	**4.21%**	**32**
Overall Averages			**12,870**	**54.2**	**84.2**	**13,830**	**91.4**	**0.7**	**13,701**	**0.78%**	**3.76%**	**4.60%**	**84**

SIPR offers two theoretical advantages over ReNet. First, it uses a greedy heuristic that follows the efficient hull of the service level-cost function, so that the system service level is considered at each step. It does this by weighing the marginal

benefit of one additional unit of each part against its associated cost and choosing the part that maximizes the benefit-to-cost ratio. With the exception of a few "inefficient" solutions that may edge out the SIPR solution, it should generally arrive at a better solution than ReNet for a given number of disassemblies. In the analysis that follows, this component of the SIPR advantage is referred to as the "heuristic advantage."

The second advantage of SIPR is that it explicitly allows for increased disassemblies to be used in place of safety levels of new parts. In most cases, this will decrease the overall cost, assuming additional cores are available. For the purposes of this analysis, this component is referred to simply as the "advantage of changing Q^d", since in some special cases it actually decreases. Table 5-3 breaks the differences in cost between SIPR and ReNet into its two components. The heuristic advantage compares the ReNet solution with the SIPR solution for the same number of disassemblies, and represents the advantage of the greedy heuristic alone. The second denotes the additional benefit of changing the number of disassemblies (Q^d). The total advantage is likewise reported, and is hereafter referred to as the "performance gap", since the difference between the two is the first topic of interest. In the rightmost column, the differences in solution time are also reported.

Several conservative conclusions can be drawn from this initial comparison. Most obvious is that SIPR always outperforms ReNet in total for the problem set. In fact, this was the case in all 48 problems (16 problems at 3 levels of uncertainty). On average, the total performance gap is 4.6%, about 0.8% of which can be attributed to the heuristic and 3.8% to the capability of increasing Q^d.

Second, while the advantage due to the heuristic itself is less than the advantage due to changing Q^d on average, it is greater in about 42% of the cases (20 of 48). Furthermore, the advantage of changing Q^d is always positive. Conversely the advantage of the heuristic is sometimes negative (3 of 48 problems). In the latter cases, ReNet found one of the inefficient solutions discussed at the start of this chapter and in Chapter 3 (p. 42-43). The overall average is also highly dependent on the product structure, as will be seen in the next section, which analyzes the main effects of the experimental factors on the performance gap in more depth.

Finally, although SIPR outperforms ReNet in all cases tested, as expected, the magnitudes of the gaps are in many cases low. In the HiHi cost-yield match cases, for example, the ReNet solutions are typically within one to two percent of the corresponding SIPR solution, even for higher levels of yield uncertainty.

Effects of Experimental Factors on Performance Gap

The following section begins with analysis and discussion regarding the main effect of the level of yield uncertainty on the performance gap. Three levels of uncertainty were tested in the experimental design, corresponding to standard deviations of the yield distributions of 0.01, 0.03, and 0.05. It should be reiterated that these correspond to the probability distributions of the yield percentages γ_p, as opposed to the actual yield quantity ψ_p, although the two are directly related. The highest level of uncertainty (st. dev. = 0.05) was chosen as the maximum value for which the tails of a normal distribution for a part with a mean yield percentage of $\gamma_p = 0.85$ (the highest in the problem set) lie within the range of 0 to 1, since by definition it is not possible to have a

yield percentage greater than 1. This ensures that only feasible quantities are included in the calculations and that effectively all of the probability is captured.

Main and Interaction Effects of Uncertainty on Performance Gap

The results of this section investigate whether the level of uncertainty has an effect on the performance gap between the ReNet and SIPR solutions. Since SIPR is a stochastic model, it is expected that it will perform better relative to ReNet as the level of uncertainty increases. The results shown in Figure 5-2 seem to support this expectation, but the range of values across the problem set indicates that more rigorous analysis is needed (Table 5-4). Note that each bar in Figure 5-2 represents an average of the performance gap across all 16 problems in the problem set for the level of uncertainty indicated.

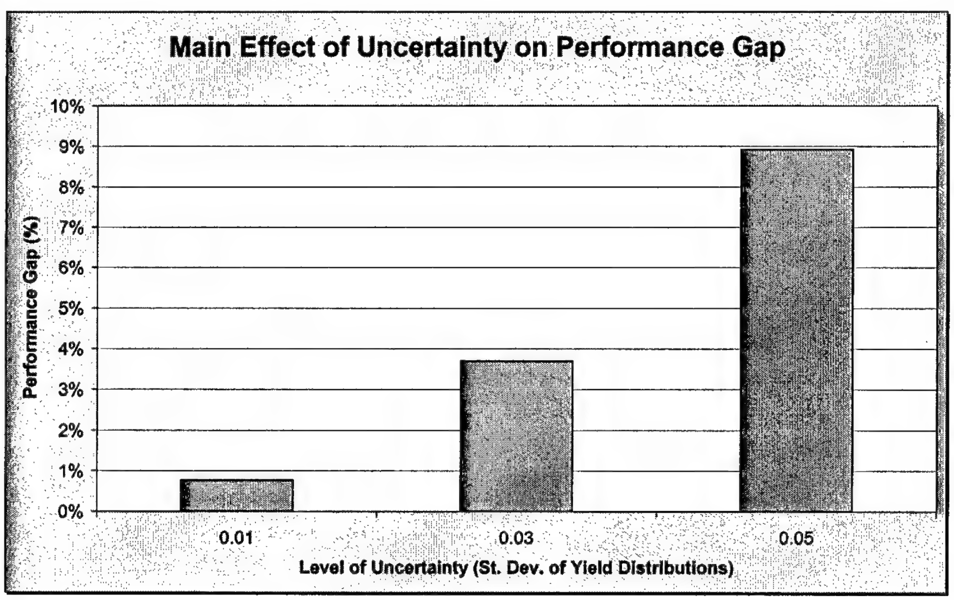


Figure 5-2: Main Effect of Uncertainty on Performance Gap

Table 5-4 shows the changes in performance gaps between the low and medium and medium and high levels of uncertainty, respectively, for each problem. It is clear from the results that the impact of the level of uncertainty on the performance gap is highly problem-specific. Although the total change in the performance gap from low to high uncertainty is positive in most cases, increasing from low to medium has a larger effect than from medium to high for the odd-numbered problems (HiHi yield-cost match). The opposite is true for the HiLo cases (even-numbered problems). This suggests that the effect of uncertainty on ReNet's performance diminishes, at least for the more common cases where the high cost parts have higher yields.

Table 5-4: Performance Gaps for Uncertainty Levels

Problem	Change in Performance Gap From...		Total Change
	Low to Medium	Medium to High	
N-B-HH-10	0.33%	-0.84%	-0.51%
N-B-HL-10	3.87%	11.95%	15.82%
N-A-HH-10	1.51%	0.04%	1.55%
N-A-HL-10	13.44%	19.75%	33.19%
W-B-HH-10	0.24%	-0.20%	0.04%
W-B-HL-10	-0.11%	3.71%	3.60%
W-A-HH-10	1.59%	-0.08%	1.50%
W-A-HL-10	4.64%	9.76%	14.40%
N-B-HH-5	0.76%	-0.60%	0.17%
N-B-HL-5	1.14%	10.38%	11.52%
N-A-HH-5	1.80%	-0.14%	1.66%
N-A-HL-5	11.73%	20.73%	32.46%
W-B-HH-5	0.46%	-0.46%	-0.01%
W-B-HL-5	-0.59%	1.92%	1.33%
W-A-HH-5	2.14%	-0.47%	1.67%
W-A-HL-5	4.04%	8.00%	12.04%
Average	2.94%	5.22%	8.15%
Standard Deviation	4.07%	7.36%	11.11%

Considering the two components of the advantage of SIPR, it is evident that the aggregate increase in the average performance gap is almost entirely attributable to the

advantage of changing Q^d (Figure 5-3a). Further, as shown in Figures 5-3b and 5-3c, the latter is significantly correlated with the cost-yield match factor. In fact, the effect of uncertainty on the performance gap is negligible for the HiHi cases (Figure 5-3b). The detailed solutions for the HiLo problems reveal the source of the bias. With increased levels of uncertainty, all eight of the HiLo problems reach a "manufacturing frontier" at either the medium or high level of uncertainty, beyond which disassemblies become more expensive than buying all new parts. In these cases, the ReNet solution for Q^d remains as before, while the SIPR solution chooses (correctly) to switch to manufacturing. Thus the advantage due to changing Q^d is actually the advantage of reducing Q^d to zero.

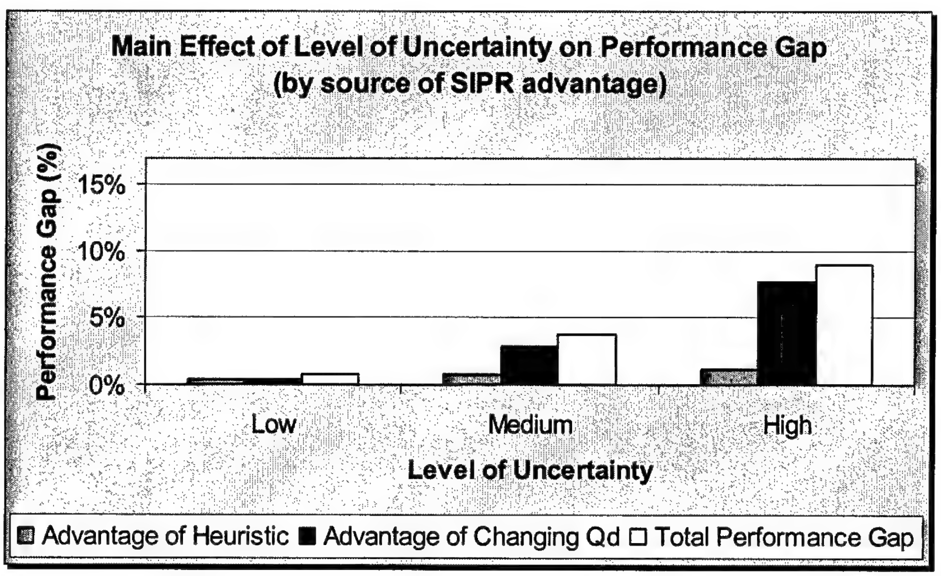


Figure 5-3a: Main Effect of Uncertainty on Performance Gap

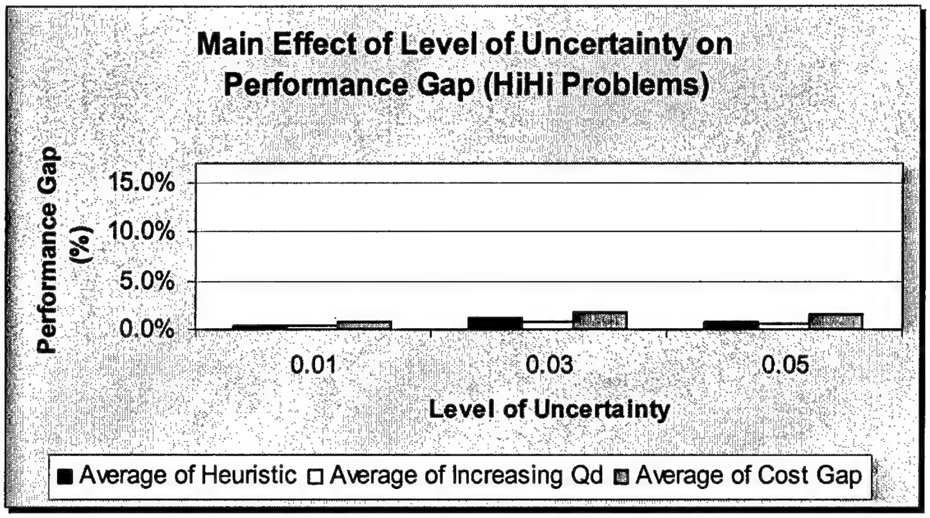


Figure 5-3b: Main Effect of Uncertainty on Performance Gap (HiHi problems)

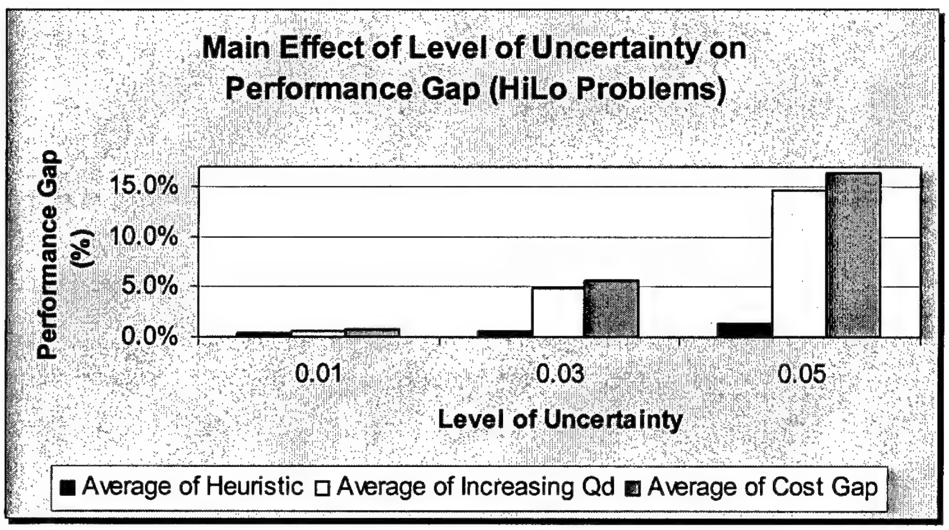


Figure 5-3c: Main Effect of Uncertainty on Performance Gap (HiLo Problems)

Main and Interaction Effects of Product Characteristics on Performance Gap

The second comparison between the SIPR and ReNet solutions centers on the product characteristic factors. Table 5-5 summarizes the main effects broken down by component, while Figure 5-4 illustrates the totals graphically.

Table 5-5: Performance Gap by Product Characteristic

Characteristic	Value	Advantage of Heuristic	Advantage of Changing Q^d	Total Performance Gap
Number of Parts	10	0.89%	3.90%	4.85%
	5	0.67%	3.34%	4.08%
Yield Range	Narrow	0.96%	5.27%	6.34%
	Wide	0.59%	1.98%	2.59%
Cost Profile	ABC	1.17%	5.25%	6.54%
	Balanced	0.38%	2.00%	2.39%
Cost-Yield Match	HiHi	0.80%	0.55%	1.35%
	HiLo	0.75%	6.70%	7.58%

Consistent with the results of the last section, the cost-yield match factor shows the greatest main effect, with the HiLo products affecting nearly an eight-percent performance gap versus just over one-percent for their HiHi counterparts. The cost profile and yield range factors also show significant main effects on the gap, although the interaction effects show that these main effects are likewise highly correlated with the cost-yield match factor. Even with the effect of the latter, the number of parts shows little main effect on the performance gap. The main effects are analyzed in more detail in the sections that follow.

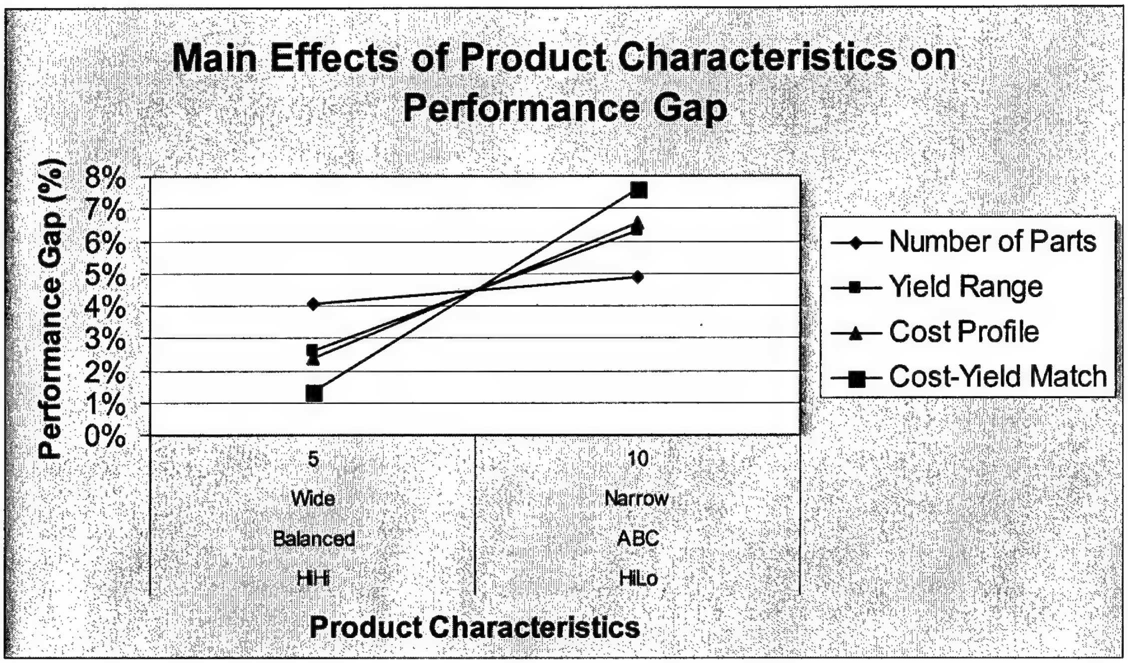


Figure 5-4: Main Effect of Product Characteristics on Performance Gap

Effect of Number of Parts. Looking at the individual problems in a pair-wise comparison, the differences between the 10-part problems and their 5-part counterparts interact strongly with the other product factors, particularly the cost-yield match (Table 5-6). For the HiHi problems only, the effect of the number of parts is negligible. The main effect for these four pairs averages about 0.25%, versus about 1.3% for the HiLo pairs.

Table 5-6: Main Effect of Number of Parts on Performance Gap (by problem)

Problem Pair	Shared Characteristics			Unique - Number of Parts		
	Yield Range	Cost Profile	Match	10 Parts	5 Parts	Main Effect
1 and 9	Narrow	Balanced	HiHi	1.27%	0.94%	0.33%
2 and 10	Narrow	Balanced	HiLo	7.33%	5.26%	2.08%
3 and 11	Narrow	ABC	HiHi	1.98%	1.71%	0.27%
4 and 12	Narrow	ABC	HiLo	16.70%	15.53%	1.18%
5 and 13	Wide	Balanced	HiHi	0.79%	0.59%	0.20%
6 and 14	Wide	Balanced	HiLo	1.72%	1.22%	0.49%
7 and 15	Wide	ABC	HiHi	1.88%	1.67%	0.21%
8 and 16	Wide	ABC	HiLo	7.14%	5.71%	1.43%
AVERAGE				4.85%	4.08%	0.77%

Effect of Yield Range. The range of yields in a product also appears to have a significant main effect on ReNet's performance (Figure 5-5a). Once again, the effect can be explained by its strong interaction with the cost-yield match factor. The average effect of yield range on the performance gap for HiHi problems is just 0.25%, about the same as the main effect of number of parts (Figure 5-5b). Its effect for HiLo problems, by contrast, is over 7.25% (Table 5-7).

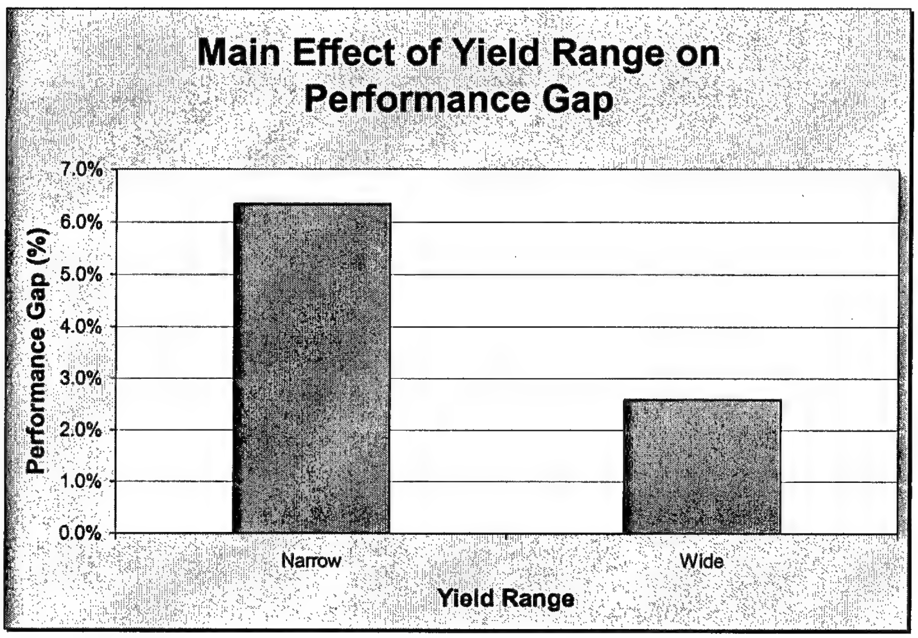


Figure 5-5a: Main Effect of Yield Range on Performance Gap

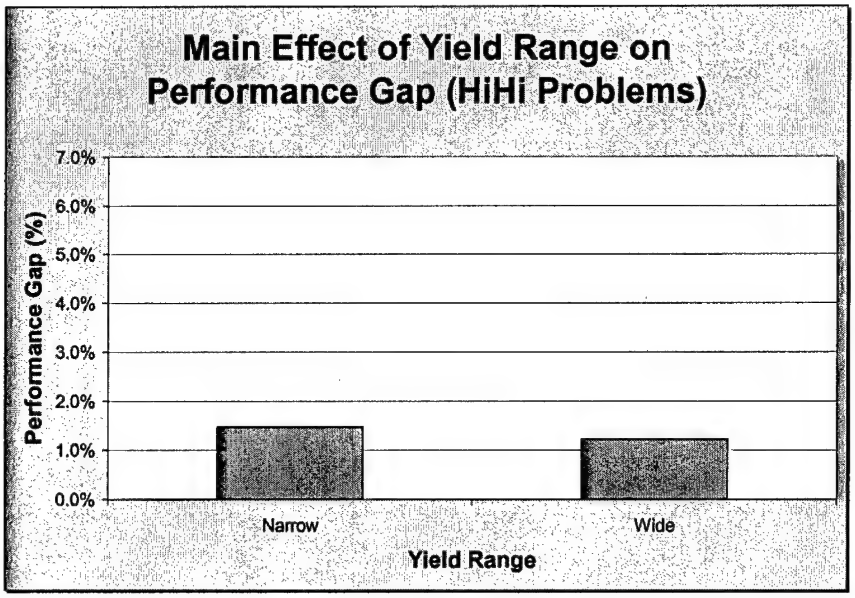
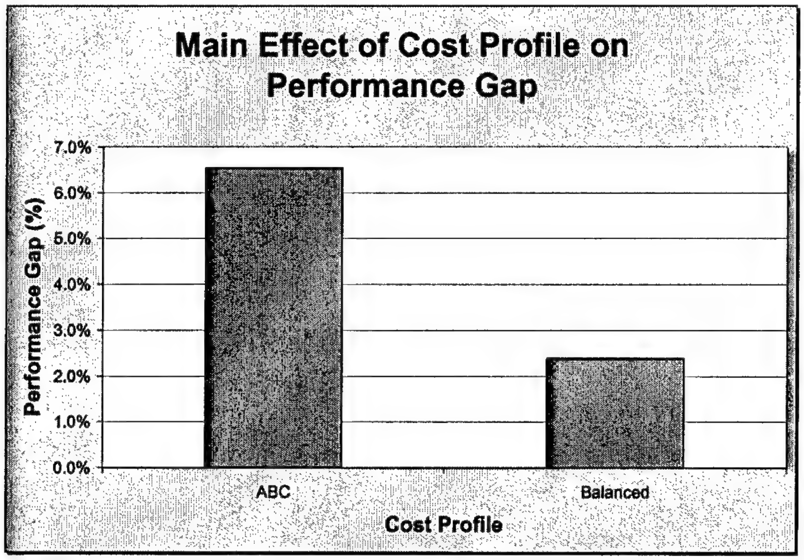


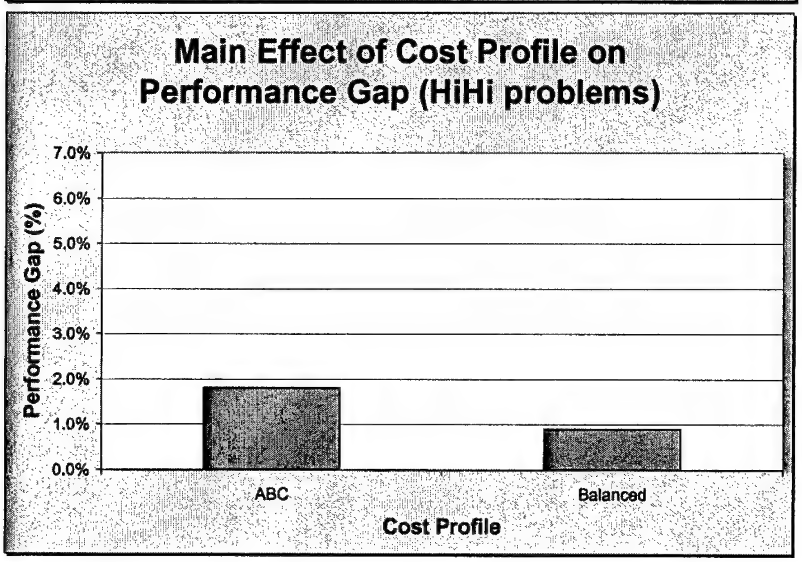
Figure 5-5b: Adjusted Main Effect of Yield Range on Performance Gap

Table 5-7: Main Effect of Yield Range on Performance Gap (by problem)

Problem Pair	Shared Characteristics			Unique - Yield Range		
	# of Parts	*Cost Profile*	*Match*	*Narrow*	*Wide*	*Main Effect*
1 and 5	10	Balanced	HiHi	1.27%	0.79%	0.48%
2 and 6	10	Balanced	HiLo	7.33%	1.72%	5.61%
3 and 7	10	ABC	HiHi	1.98%	1.88%	0.11%
4 and 8	10	ABC	HiLo	16.70%	7.14%	9.56%
9 and 13	5	Balanced	HiHi	0.94%	0.59%	0.35%
10 and 14	5	Balanced	HiLo	5.26%	1.22%	4.03%
11 and 15	5	ABC	HiHi	1.71%	1.67%	0.05%
12 and 16	5	ABC	HiLo	15.53%	5.71%	9.82%
AVERAGE				6.34%	2.59%	3.75%

Effect of Cost Profile. Not surprisingly, the effect of the cost profile on the performance gap, shown graphically in Figure 5-6a, interacts strongly with the cost-yield match factor. In this case, however, the main effect is relatively stronger even after blocking the interaction effect, about 0.9%. The practical significance of this effect, of course, would depend on the dollar values involved in an operational setting.





Figures 5-6: (a) Main Effects of Cost Profile on Performance Gap
(b) Adjusted Main Effects of Cost Profile on Performance Gap

Table 5-8: Main Effect of Cost Profile on Performance Gap (by problem)

Problem Pair	Shared Characteristics			Unique - Cost Profile		
	# of Parts	Yield Range	Match	ABC	Balanced	Main Effect
1 and 3	10	Narrow	HiHi	1.98%	1.27%	0.72%
2 and 4	10	Narrow	HiLo	16.70%	7.33%	9.37%
5 and 7	10	Wide	HiHi	1.88%	0.79%	1.08%
6 and 8	10	Wide	HiLo	7.14%	1.72%	5.42%
9 and 11	5	Narrow	HiHi	1.71%	0.94%	0.78%
10 and 12	5	Narrow	HiLo	15.53%	5.26%	10.27%
13 and 15	5	Wide	HiHi	1.67%	0.59%	1.08%
14 and 16	5	Wide	HiLo	5.71%	1.22%	4.48%
AVERAGE				6.54%	2.39%	4.15%

Effect of Cost-Yield Match. The final product characteristic is the cost-yield match, which has already been discussed at length since it interacts so strongly with the other factors. Figure 5-7 illustrates the magnitude of its main effect. The average gap of 7.6% for the HiLo problems is more than five-times that of the HiHi problems.

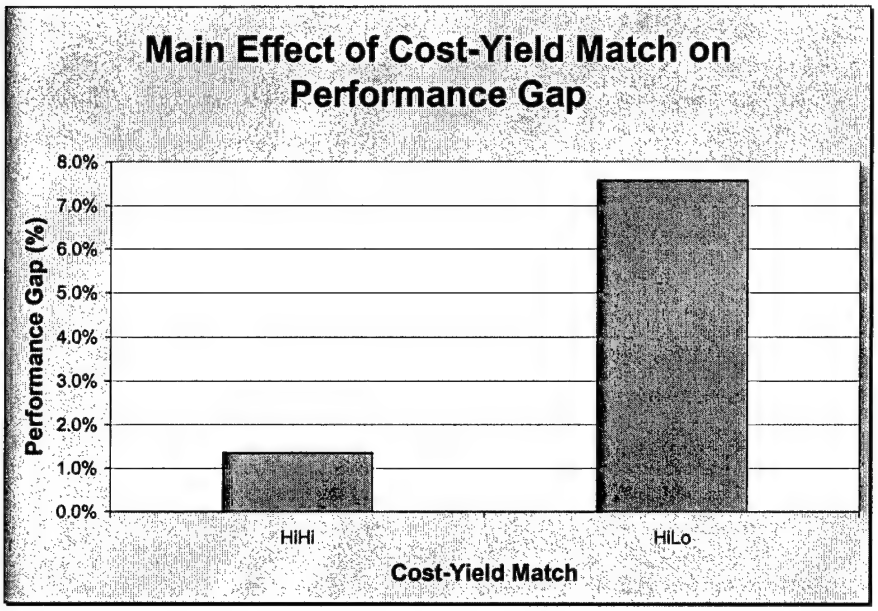


Figure 5-7: Main Effect of Cost-Yield Match on Performance Gap

Table 5-9: Main Effect of Cost-Yield Match on Performance Gap (by problem)

Problem Pair	Shared Characteristics			Unique - Cost-Yield Match		
	# of Parts	Yield Range	Cost Profile	HiLo	HiHi	Main Effect
1 and 2	10	Narrow	Balanced	7.33%	1.27%	6.06%
3 and 4	10	Narrow	ABC	16.70%	1.98%	14.72%
5 and 6	10	Wide	Balanced	1.72%	0.79%	0.93%
7 and 8	10	Wide	ABC	7.14%	1.88%	5.27%
9 and 10	5	Narrow	Balanced	5.26%	0.94%	4.32%
11 and 12	5	Narrow	ABC	15.53%	1.71%	13.82%
13 and 14	5	Wide	Balanced	1.22%	0.59%	0.63%
15 and 16	5	Wide	ABC	5.71%	1.67%	4.04%
AVERAGE				7.58%	1.35%	6.22%

Implications of the Results of Research Question 1

Based on the results reported in this section, it is evident that the effects of uncertainty and the product structure factors affect ReNet's overall performance to varying degrees. The cost-yield match factor showed the most dramatic effect, and also had a strong interaction with all other factors, as expected.

Two conclusions can be drawn from the results that help to answer the first research question. First, given its computational simplicity, ReNet performs well relative to SIPR for HiHi problems, generating solutions that are within 1.35% of the SIPR solutions on average. By contrast, ReNet performs very poorly for HiLo problems, with solutions that are 7.6% higher than SIPR on average. The only other factor that showed practical significance after adjusting for the cost-yield match interaction was the cost profile, with the gap widening by about 1% for ABC problems over those with a balanced cost profile. This is consistent with expectations, since the solution for ABC problems is largely driven by the "significant few" high-cost parts. Whereas the sorting heuristic in ReNet gives it the potential to avoid an additional unit or two of the high-cost item, SIPR will typically order much higher levels of the lower-cost items to avoid the high cost parts. This is an inherent quantitative trade-off of the ReNet methodology.

The second conclusion, which was briefly noted in Chapter 3 (p. 35-36), is that ReNet fails to detect the point at which a problem crosses the manufacturing frontier, where it is no longer cost-effective to disassemble cores. In fact, as was also noted in Chapter 3 (p. 35), an additional check procedure had to be included in SIPR to preclude the same thing from happening. Despite its limitations, however, ReNet performed quite

well relative to SIPR, particularly considering the fact that the HiLo problems would rarely be seen in a viable remanufacturing setting due to their prohibitive cost.

Research Question 2

What effect does the reduction of yield uncertainty (i.e. prior knowledge of expected yields) have on cost?

In the preceding section, results were presented indicating that increased uncertainty had very little effect on the performance of ReNet relative to that of SIPR, at least for the more common HiHi problems. In this section, the magnitude of its effect on solution cost is presented and discussed. The results from this point forward use only the optimal/near-optimal solution costs of the SIPR methodology. Since this research question hinges on the effects of a single factor, the main and interaction effects are discussed separately.

Main Effect of Uncertainty on Cost

Figure 5-8 and Table 5-10 show the aggregate results of the comparison of solution costs for each level of uncertainty tested, broken down by cost component. Each bar represents an average cost across all 16 problems in the set for the given level of uncertainty. The results are consistent with expectations, in that the average solution cost increases with the level of uncertainty. The average cost increases by 5.8% from a standard deviation of 0.01 to 0.03, and by 1.8% from a standard deviation of 0.03 to 0.05. The core purchase and disassembly costs decrease at a decreasing rate with the level of uncertainty, while the new part purchase cost increases at a decreasing rate. Based on the results associated with the first research question, however, caution must be exercised

when analyzing the main effects due to the wide range of values across the problem set. To that end, the interaction effects are explored in more detail in the next section.

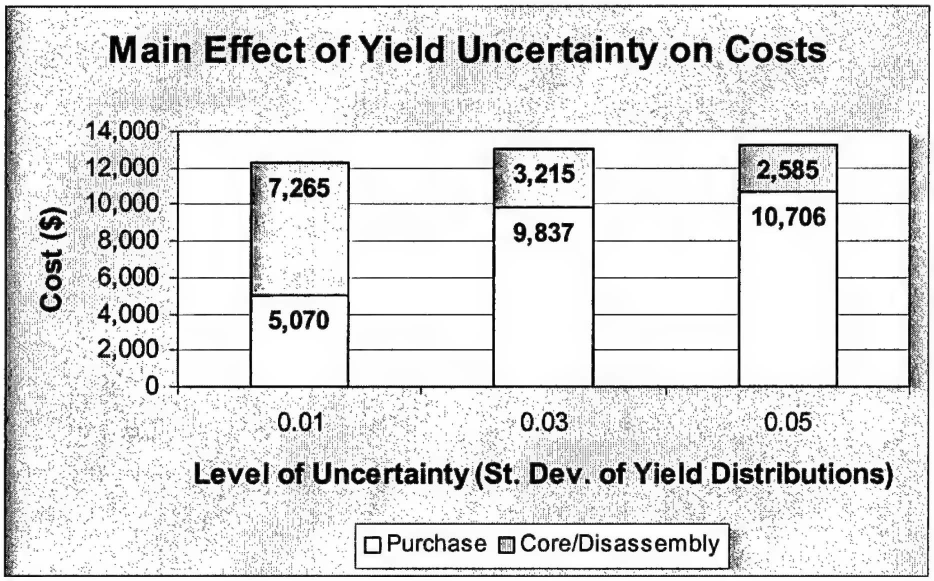


Figure 5-8: Main Effect of Yield Uncertainty on Cost

Table 5-10: Main Effect of Yield Uncertainty on Cost (by Component)

Cost Component	Level of Uncertainty		
	0.01	0.03	0.05
Holding	0.54	0.65	0.85
Core Purchase/Disassembly	7,265.00	3,215.00	2,585.00
New Part Purchase	5,069.72	9,836.80	10,705.66
Total	12,335.26	13,052.45	13,291.52

Two-Way Interactions of Uncertainty Effect

Figures 5-9 through 5-12 show the effects of the two-way interactions between the level of uncertainty and each product characteristic. These second-order effects are quite small in all cases but one. Once again, the cost-yield match factor interacts strongly with the uncertainty factor. Recall from earlier discussion that all of the HiLo problems crossed the manufacturing frontier at either the medium or high level of uncertainty. At this point, the cost effectively flattens out at the cost of buying 50 complete sets of parts with no disassemblies. Therefore beyond this point, yield uncertainty has no effect. This effect can be seen in both the main effect of uncertainty, which increases at a decreasing rate beyond the medium uncertainty level, and in its interaction effects (Figures 5-9 through 5-12), where the same is true. The one exception is shown in Figure 5-12, where the effect of uncertainty on cost steadily increases for HiHi parts, a result that is more consistent with expectations.

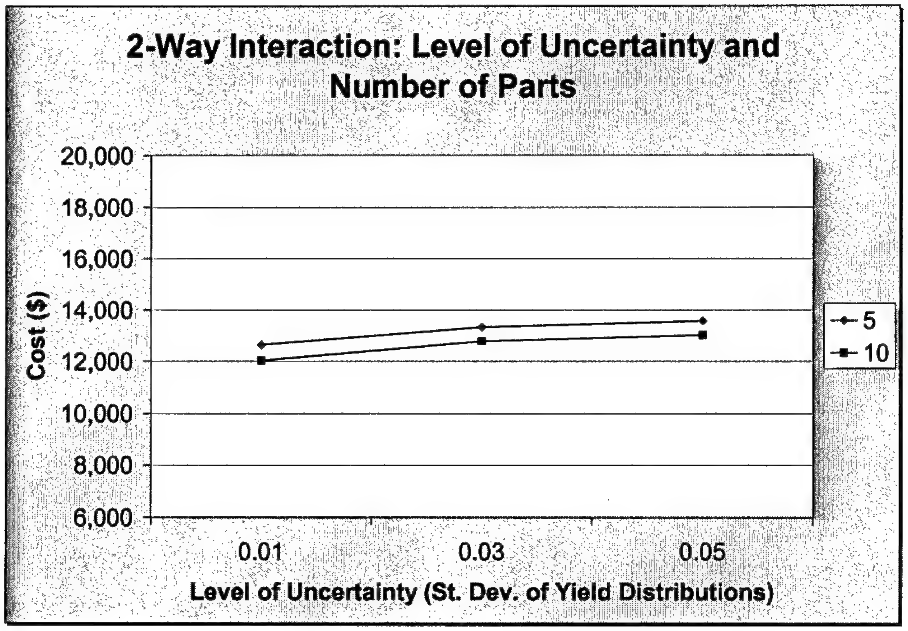


Figure 5-9: Two-Way Interaction Between Level of Uncertainty and Number of Parts

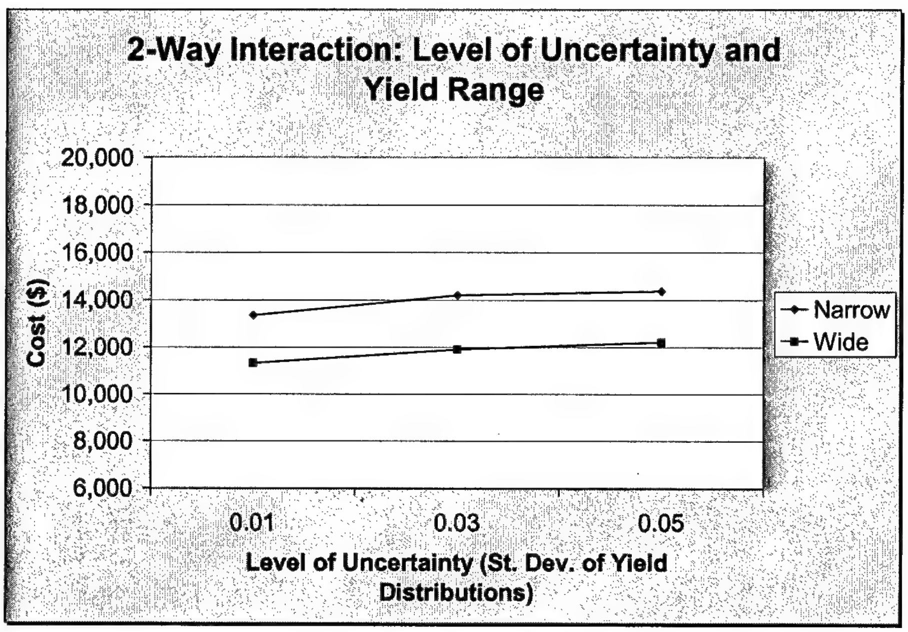


Figure 5-10: Two-Way Interaction Between Level of Uncertainty and Yield Range

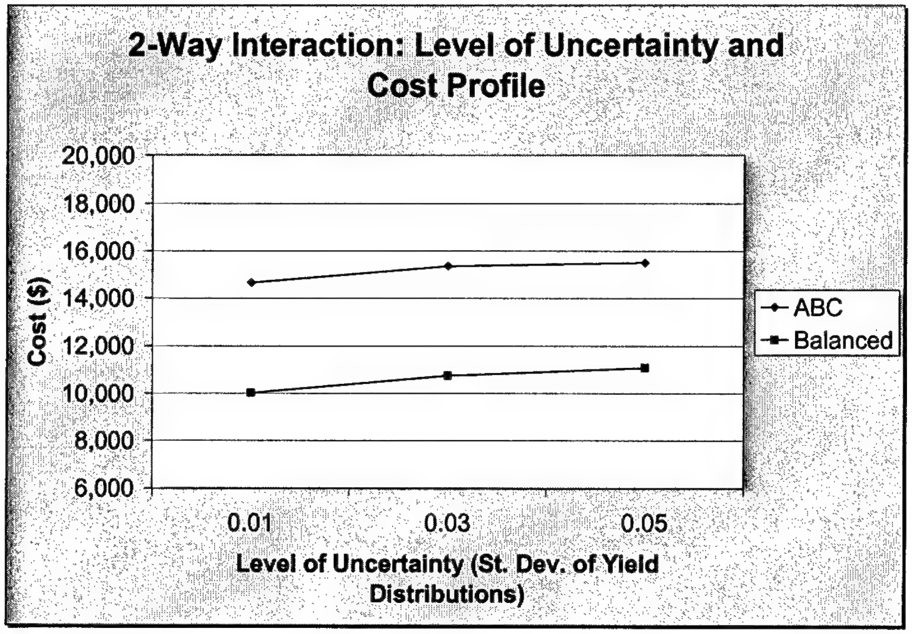


Figure 5-11: Two-Way Interaction Between Level of Uncertainty and Cost Profile

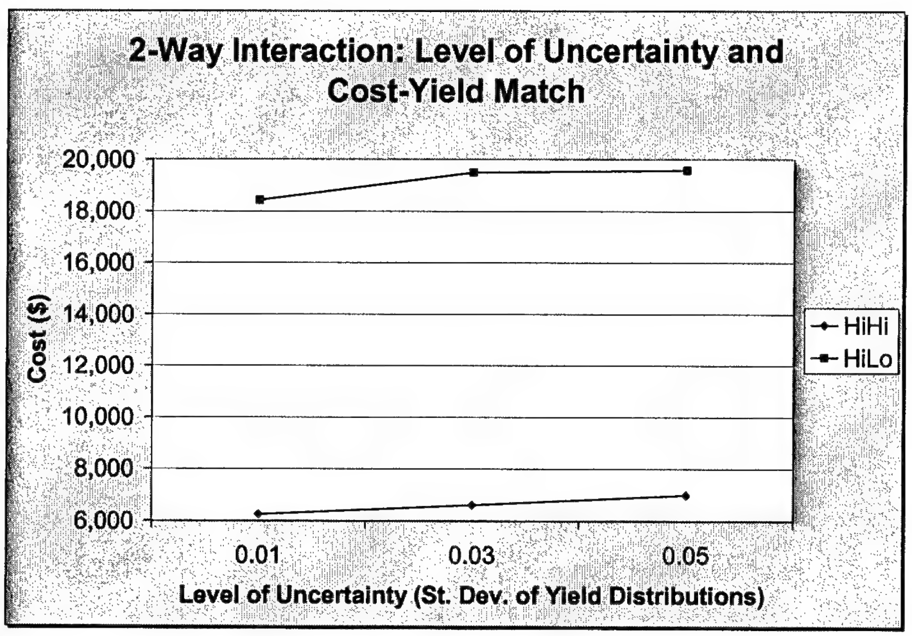


Figure 5-12: Two-Way Interaction Between Level of Uncertainty and Cost-Yield Match

In total, the interactions illustrated in Figures 5-9 through 5-12 indicate that the HiLo problems under the highest level of uncertainty, which are no longer affected because their costs have leveled, are pulling down the aggregate effects of uncertainty on the other factors. Blocking the effects of the HiLo products, the results fall in line with expectations. As Figure 5-13 shows, the average cost of the solution now increases steadily with increased uncertainty, as originally expected. The cost components do not increase proportionally, however, as the core purchase and disassembly costs begin to decrease from the moderate to the high level of uncertainty. This indicates that the disassembly option becomes less attractive as the yield uncertainty increases, leading to higher requirements for new parts.

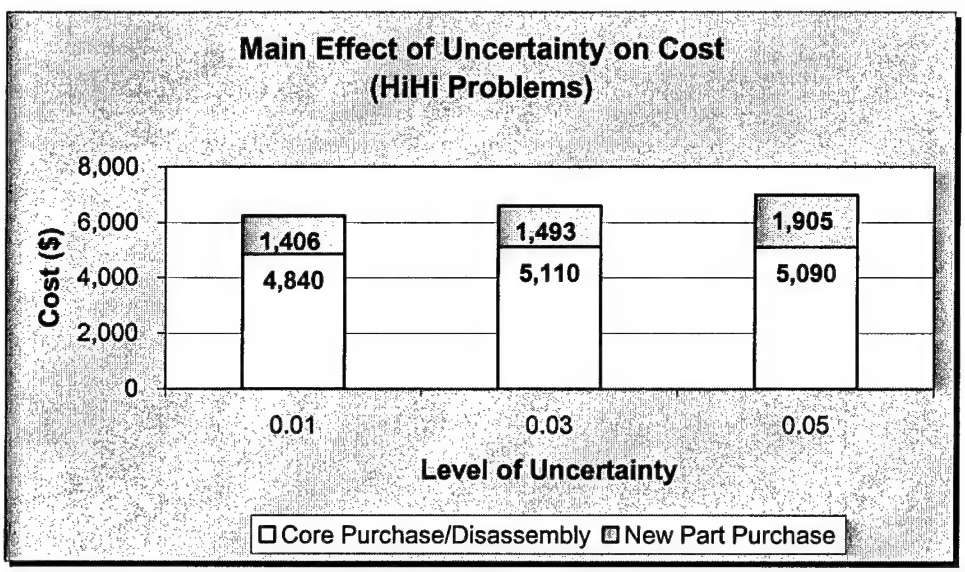


Figure 5-13: Adjusted Main Effect of Uncertainty on Cost (HiHi problems only)

Table 5-11: Main Effects of Uncertainty on Cost (HiHi problems only)

Cost Component	Level of Uncertainty		
	0.01	0.03	0.05
Holding	0.36	1.03	1.67
Core Purchase/Disassembly	4,840.00	5,110.00	5,090.00
New Part Purchase	1,406.03	1,492.83	1,905.08
Total	6,246.39	6,603.86	6,996.75

Similarly, the interaction effects between level of uncertainty and the product characteristics (Figures 5-14 through 5-16) are much more consistent with expectations when the confounding effect of the cost-yield match factor is blocked.

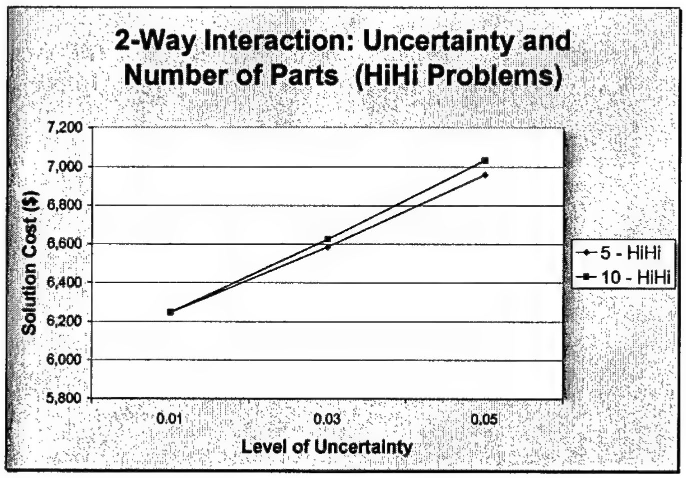


Figure 5-14: Interaction Effect – Uncertainty & Number of Parts

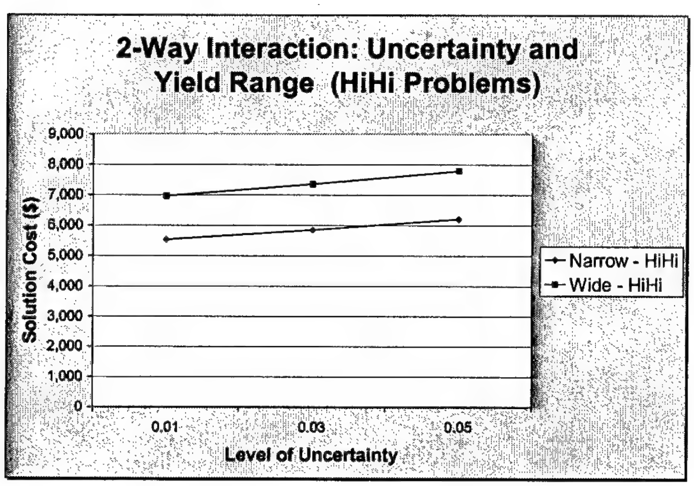


Figure 5-15: Interaction Effect – Uncertainty & Yield Range

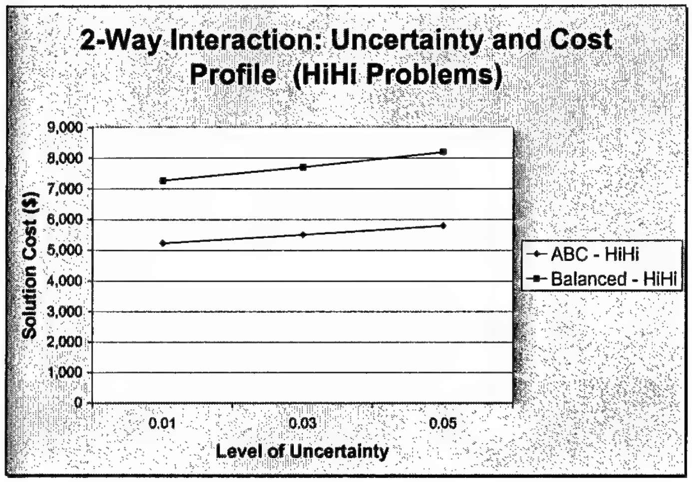


Figure 5-16: Interaction Effect – Uncertainty & Cost Profile

In all adjusted cases, increased uncertainty steadily increases the cost of the solution. The most significant interaction is found between uncertainty and the number of parts. At the lowest level of uncertainty, the average solution cost for 5- and 10-part problems is nearly equal. However the gap increases steadily as the level of uncertainty grows, indicating that the complexity of a product (in terms of the number of parts) can amplify the uncertainty effects, as should be expected. Keeping in mind that even the 10-part problem represents a rather benign case, the practical impact of the number of parts could be substantial in a real product with hundreds of parts.

<u>Implications of the Results of Research Question 2</u>

The first notable implication of the results of this section relates to the potential effects of the high degree of correlation between the product characteristics and the results. Great care must be taken in drawing conclusions, particularly when extreme cases are included, as a few problems can easily have a confounding effect on the rest.

With regard to the research question, the results indicate that the effect of uncertainty indeed has a significant impact on cost, as suspected. This lends support to the argument that remanufacturers, particularly OEMs that have complete control over products throughout their life cycle, pay particularly close attention to the reduction of uncertainty in yields. This attention can be focused at any of several points in the product life cycle. First, design of new products should take reliability characteristics into consideration to a higher degree, since decreased reliability not only affects current new product sales but, as was shown here, can also have a dramatic effect on the cost of remanufacturing products later.

A second potential area for improvement is in the forecasting of yields. Low-cost options like tracking the age and usage of cores may provide improvements to yield forecasts that have a significant impact on remanufacturing costs. Likewise, capital investments in diagnostic equipment and/or installed computers that track detailed usage characteristics may be able to provide a level of detail that allows remanufacturing managers to more accurately determine yields in advance.

The final implication of the results reported here is that the effect of uncertainty grows with the complexity of the product, as determined by its number of parts. By definition, increased yield uncertainty requires additional safety stock for each part. A greater number of parts should therefore be expected to require a proportional increase in total safety stock across all items. This expectation was confirmed in the results, and suggests that remanufacturers of complex products need to be much more cognizant of the uncertainty of yields than do remanufacturers of simpler products with relatively fewer parts.

Research Question 3

How do a product's structural characteristics (number of parts, range of yield percentages, part cost profile, and cost-yield match) affect the cost of the solution?

The structural characteristics of a product, as already noted throughout the previous sections, have a dramatic effect on the cost of the solution. As shown in Figure 5-17, the cost-yield match characteristic has the most profound impact, with the cost increasing by over 189% on average from a High-High match (HiHi) to a High-Low match (HiLo). The yield range and cost profile also have a significant impact, representing cost increases of 18% and 43%, respectively, from wide to narrow and from balanced to ABC. The number of parts shows the least main effect on average, with just a 4.5% cost increase from 10 parts to 5 parts. Once again, it is clear from the wide range of values and skewed distributions that the averages do not adequately explain the effects of product characteristics. It is also likely that the cost-yield match factor will once again show a significant interaction effect with the other factors, as in the previous two sections. The individual effects are discussed in more detail in the sections that follow.

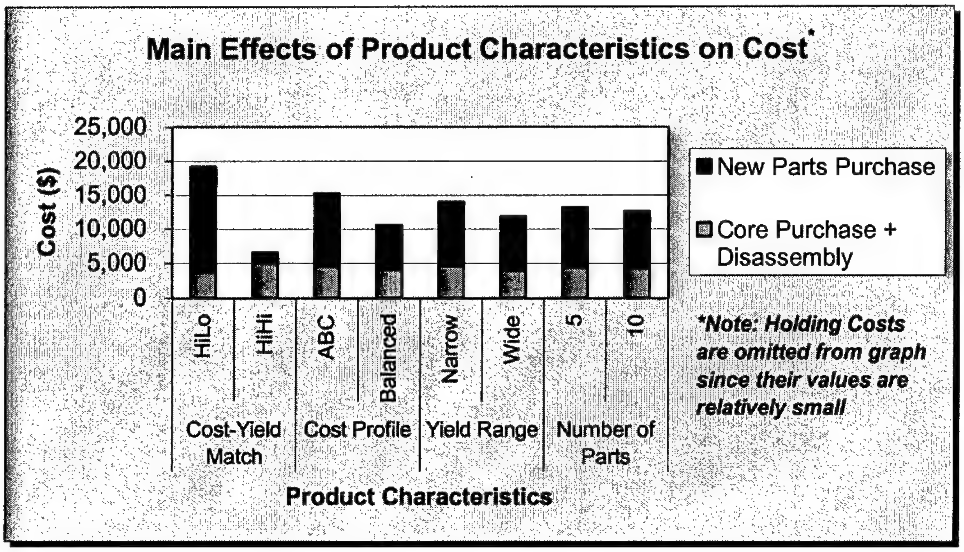


Figure 5-17: Main Effects of Product Characteristics on Solution Cost

<u>Main and Interaction Effects of Product Characteristics on Solution Cost</u>

Number of Parts

The slightly higher average cost across the 5-part problems may be appear to be inconsistent with expectations at first glance, given the general rule that increased complexity leads to increased cost. In fact, the results of the first two research questions display that very behavior. Increasing from 5 to 10 parts degraded ReNet's performance slightly in the first section, while it increased the effect of yield uncertainty on cost in the second. In the case of the effect of number of parts on solution cost the reverse is true, with 5-part problems showing an average cost $570 higher than the 10-part problems. Table 5-12 illuminates the source of the difference. Once again, the cost-yield match factor drives the difference. In this case, however, the cost-yield match factor interacts with the cost profile factor somewhat. The balanced-HiLo problems show a higher cost for the 10-part problems, while the ABC-HiLo problems show a much higher cost for the

5-part problems. Since the magnitude is so much greater for the latter, these problems dominate the results. In fact, the main effect of the number of parts is reduced to just $77 when these two pairs are removed from the analysis.

Table 5-12: Effect of Number of Parts on Solution Cost

Problem Pair	Shared Characteristics			Unique - Number of Parts		
	Yield Range	Cost Profile	Match	5	10	Main Effect
1 and 9	Narrow	Balanced	HiHi	6,435	6,345	91
2 and 10	Narrow	Balanced	HiLo	16,520	16,820	-299
3 and 11	Narrow	ABC	HiHi	5,275	5,387	-112
4 and 12	Narrow	ABC	HiLo	28,462	26,605	1,857
5 and 13	Wide	Balanced	HiHi	9,128	8,978	150
6 and 14	Wide	Balanced	HiLo	10,336	10,349	-13
7 and 15	Wide	ABC	HiHi	5,547	5,831	-284
8 and 16	Wide	ABC	HiLo	23,720	20,552	3,169
AVERAGE				11,714	11,208	570

The specific Q^d and $\{Q_p\}$ solutions of the ABC-HiLo pairs shed more light on the reason for the large effect of the number of parts. The high cost – low yield part(s) drive the optimal number of disassemblies to extremely high quantities for these cases, since many more cores must be disassembled to get enough of the high-cost parts to meet demand. This has the effect of creating a great deal of excess inventory for the higher-yield parts, most of which reach a service level of 100% and therefore offer no additional benefit in reducing the order quantity of the high-cost, low-yield parts. In the 5-part cases, this left only one or two parts whose order quantities could be increased to reduce the order quantities of the cost-driver. For the corresponding 10-part cases, four or five parts were still below their target service levels at the optimal Q^d, which offered much more flexibility in finding a more efficient solution.

In addition, at higher levels of uncertainty, these products crossed into the "manufacturing" region, where no cores were disassembled. Beyond this point, the

controlled experimental property of equal expected values of cores across all problems becomes moot, since no cores are being disassembled. The difference in these cases is strictly the total cost of new parts for each unit, which in this case was slightly higher for the 5-part problems than for the 10-part problems. Although the latter is a by-product of the parameters of the experimental problem set, the results for the low-uncertainty cases indicate that the cost would likely have been higher in these cases even had it still been economically beneficial to disassemble cores.

Yield Range

The main effect of the yield range on cost was nearly $2200, with the narrow problems having the higher cost on average. In this case, the detailed results in Table 5-13 indicate the existence of higher-order interactions, and so graphs of the two-way interactions would be difficult to interpret. Figure 5-18 shows the three-way interaction effects graphically for the interaction between yield range and the cost profile and cost-yield match factors. The number of parts was excluded, since it shows little effect on the interactions with the other three.

Table 5-13: Effect of Yield Range on Solution Cost

Problem Pair	Shared Characteristics			Unique - Yield Range		
	Number of Parts	Cost Profile	Match	Narrow	Wide	Main Effect
1 and 5	10	Balanced	HiHi	6,345	8,978	-2,633
2 and 6	10	Balanced	HiLo	16,820	10,349	6,471
3 and 7	10	ABC	HiHi	5,387	5,831	-444
4 and 8	10	ABC	HiLo	26,605	20,552	6,054
9 and 13	5	Balanced	HiHi	6,435	9,128	-2,693
10 and 14	5	Balanced	HiLo	16,520	10,336	6,185
11 and 15	5	ABC	HiHi	5,275	5,547	-272
12 and 16	5	ABC	HiLo	28,462	23,720	4,742
AVERAGE				13,981	11,805	2,176

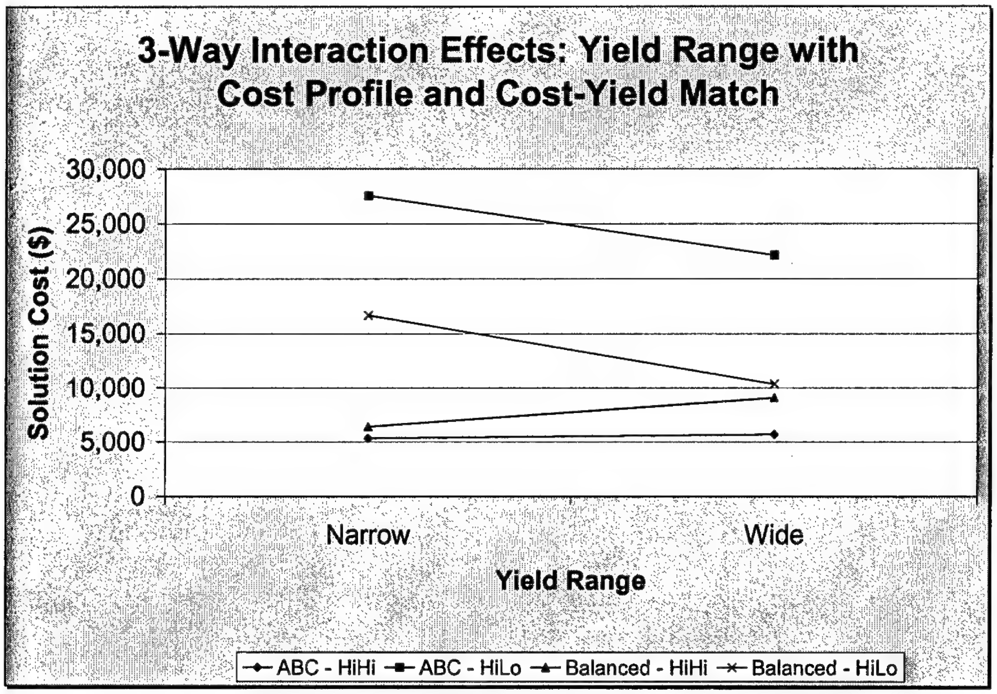


Figure 5-18: Three-Way Interaction Effects of Yield Range, Cost Profile, and Cost-Yield Match on Solution Cost

The first notable interaction in Figure 5-18 is the opposite effect of yield range on the HiHi and HiLo problems. A narrow yield range has a much larger solution cost for the HiLo problems, while the reverse is true for the HiHi problems, albeit to a lesser degree. This makes sense, considering that the lowest yield is 0.15 for all of the HiLo problems, but the remaining yields are much lower on average for the narrow range than for the wide range. This will obviously increase the cost, since more parts must be ordered to account for the lower yields. For the HiHi problems, the highest yield is always 0.85, but the remaining yields are higher on average for the narrow case, which has the opposite effect of decreasing the cost. This interaction is worse for the balanced cases, since the costs associated with these "remaining yields" are higher relative to the highest-cost part.

The other notable result is that the yield range has very little effect on ABC-HiHi products. This logically follows from the fact that the impact of the "insignificant many" parts is very small relative to the high-cost part or parts. This holds true regardless of their yield range, since it is driven almost entirely by the cost structure of the problem, not the yields. That said, the wide-range does have a slightly larger cost, as would be expected, but its magnitude is very small relative to the others.

Cost Profile

As in the last case, the effects of cost profile on cost show significant interaction effects, particularly between cost profile and cost-yield match (Table 5-14). The yield range factor also has an interaction effect, in the form of amplifying the interaction effect between cost profile and cost-yield match. Since these are precisely the same interactions noted in the previous section, there is no need to re-analyze the higher-order interaction effects.

Table 5-14: Effect of Cost Profile on Solution Cost

Problem Pair	Shared Characteristics			Unique - Cost Profile		
	Number of Parts	Yield Range	Match	ABC	Balanced	Main Effect
1 and 3	10	Narrow	HiHi	5,387	6,345	-958
2 and 4	10	Narrow	HiLo	26,605	16,820	9,786
5 and 7	10	Wide	HiHi	5,831	8,978	-3,147
6 and 8	10	Wide	HiLo	20,552	10,349	·10,203
9 and 11	5	Narrow	HiHi	5,275	6,435	-1,161
10 and 12	5	Narrow	HiLo	28,462	16,520	11,942
13 and 15	5	Wide	HiHi	5,547	9,128	-3,582
14 and 16	5	Wide	HiLo	23,720	10,336	13,385
AVERAGE				15,172	10,614	4,559

Cost-Yield Match

The final effect to examine is that of the cost-yield match factor, which has been shown to have significant interaction effects with almost every other factor in the study. The most noteworthy, yet by now least surprising, result is that the HiLo products result in a higher solution cost in all cases (Table 5-15). Since this factor has the most significant impact on the solution cost, as well as on the effects of other factors, it largely forms the basis for the general implications of the product structure characteristics with regard to cost. These are discussed in the following section.

Table 5-15: Effect of Cost-Yield Match on Solution Cost

Problem Pair	Shared Characteristics			Unique - Cost-Yield Match		
	Number of Parts	Yield Range	Cost Profile	HiLo	HiHi	Main Effect
1 and 2	10	Narrow	Balanced	16,820	6,345	10,475
3 and 4	10	Narrow	ABC	26,605	5,387	21,219
5 and 6	10	Wide	Balanced	10,349	8,978	1,371
7 and 8	10	Wide	ABC	20,552	5,831	14,721
9 and 10	5	Narrow	Balanced	16,520	6,435	10,085
11 and 12	5	Narrow	ABC	28,462	5,275	23,188
13 and 14	5	Wide	Balanced	10,336	9,128	1,207
15 and 16	5	Wide	ABC	23,720	5,547	18,174
AVERAGE				19,170	6,616	12,555

Implications of the Results of Research Question 3

In an effort to answer research question 3, this section has provided a detailed look at the main and interaction effects of the four product characteristics on solution cost. Table 5-16 summarizes the effects according to the three that showed the most significant main and interaction effects: yield range, cost profile, and cost-yield match. The number of parts is omitted here to clarify the discussion, since the relative order of the solution costs is the same for both 10- and 5-part problems. It should be noted first

that the values reported in Table 5-16 are the averages across both levels of the number of parts factor and all three levels of uncertainty, so each represents an average of six runs.

Table 5-16: Summary of Effects of Product Characteristics on Solution Cost (each cost is an average of 6 runs representing 5 and 10 part problems across 3 levels of uncertainty)

Yield Range	Cost Profile	Cost-Yield Match	Solution Cost
Narrow	ABC	HiHi	5,331
Wide	ABC	HiHi	5,689
Narrow	Balanced	HiHi	6,390
Wide	Balanced	HiHi	9,053
Wide	Balanced	HiLo	10,342
Narrow	Balanced	HiLo	16,670
Wide	ABC	HiLo	22,136
Narrow	ABC	HiLo	27,534

The results can be summed up by the following rule of thumb. If the high-yield part has the highest cost, the solution cost will be very low, and if the low-yield part has the highest cost, the solution cost will be very high. Although this rule applies for any cost profile, yield range, or number of parts, the degree of solution "goodness" (for HiHi problems) and of solution "badness" (for HiLo problems) is amplified by an ABC cost profile, a narrow yield range, and/or a fewer number of parts. In fact, in terms of the sets of product characteristics, the top (HiHi) and bottom (HiLo) of Table 5-14 are mirror images. It therefore offers a very clean and consistent guide to the discussion of implications that follows.

The first implication of the results of this research question should be clear. The greatest effort and most resources should be focused on increasing the yields of the high cost parts. Whether it means increased attention to reliability engineering in the design phases of a product's life, more frequent and rigorous preventative maintenance, improvements in quality control, or a combination of the above largely depends on one's

perspective and position, but the impact of the cost-yield match factor is indisputably the single greatest determinant of solution cost.

A second implication, while related to the first, takes a more strategic perspective. The results suggest that, barring the ability to transform a product toward a HiHi/ABC structure as described above, perhaps remanufacturing of the product may not be economically viable. In fact, as product lines age and the average yields of expensive parts decrease, the strategic decision of whether or not to continue remanufacturing could very well change over time.

Finally, assuming a product has a HiHi cost-yield match, there is some benefit to trying to homogenize the yields. In other words, the more clustered the average yields are in a product, the lower the cost (i.e. the narrow yield range factor-level combination). If the cost profile is more balanced than ABC in nature, this effect is even more critical. These implications are generalizable in the sense that three of the product characteristics isolated in this section included the most extreme values possible.

Conclusions

This chapter began with the results of two pilot studies conducted prior to the main and secondary experiments. The first showed that a simple, computationally efficient local search heuristic (Cheapest Final Step) did an adequate job of finding better solutions that SIPR may have missed in its search. A more rigorous (and computationally expensive) heuristic offered little additional improvement. The second pilot study compared the SIPR solutions to the corresponding optima for a sample set of 384 problems, and demonstrated that SIPR (with the added CFS heuristic) found the

optimal solution in 96% of the cases, thus empirically supporting the near-optimality of the SIPR methodology.

The results of the main experiment were then presented, and the three research questions were answered in light of the results. The first showed that ReNet, while much faster, found solutions very close to the near-optimal solutions of SIPR. The one exception was found in the problems with a HiLo cost-yield match, for which ReNet failed to account for the phenomenon of the "manufacturing frontier." SIPR therefore performed much better relative to ReNet for these cases, but they were shown throughout the chapter to be unlikely candidates for a profitable remanufacturing operation.

The second research question quantified the effects of uncertainty on the cost of the solution. The results suggest that increased levels of yield uncertainty lead to significantly higher costs, as expected. Once the confounding effects of the cost-yield match factor were blocked, it also became apparent that a greater number of parts amplifies the effect of yield uncertainty on cost.

The third and final research question was answered next, and showed the effects of product characteristics on solution cost. The results indicate that the HiLo cost-yield match indicates a poor match for a profitable remanufacturing operation. They further suggest that an ABC cost profile is preferable to a balanced cost profile with respect to cost, and that a narrower yield range further reduces cost.

Still, in an experimental setting such as this certain factors remained fixed that may have affected the results reported to this point. In order to have a higher degree of confidence in the generalizability of the results, Chapter 6 reports the results of four sensitivity experiments that were conducted to that end. In addition, the results of a

multi-period experiment are reported, with the demand pattern added as an experimental factor.

CHAPTER 6

SENSITIVITY ANALYSIS AND SECONDARY EXPERIMENTAL RESULTS

Chapter 5 presented the results of the main experiment, in which several factors were held at fixed levels based on real examples in industry. This was done to focus the experiment on those factors thought to have the greatest impact. This chapter presents the results of a set of sensitivity analyses that explores the behavior of the single-period problem under more rigorous conditions, in order to determine whether or not the fixed factors masked any significant problem characteristics. Also presented in this chapter are the results of a limited multi-period experiment, designed to test the performance of the models for this more difficult case. The sensitivity analyses are discussed first.

Sensitivity Analysis

This section presents the results of four sensitivity analyses that were conducted to strengthen the results of the main experiment. Three factors are varied that were fixed in the main experiment, but were expected to have an impact on solution cost: core purchase/disassembly cost, target service level, and end-item demand. Pilot runs were conducted to determine a broad enough range of values for each variable so that relevant differences could be detected. In a fourth sensitivity test, the level of uncertainty is varied across a much broader range than that of the main experiment. The latter analysis

required the selection of narrow-range problems, so that the longer tails of the normal distribution could be kept within the feasible range, as in the main experiment.

For the purpose of the sensitivity runs, a subset of four problems was selected. All were 10-part problems, since the number of parts factor showed very little effect on solution cost and its interactions, where present, were straightforward and consistent with expectations. The problems were also chosen from the more realistic set of HiHi problems, in order to avoid the complex and often veiling effect of the HiLo problems. As previously discussed, HiLo problems would rarely be seen in a viable remanufacturing operation due to their prohibitive cost and structure. Two of the problems tested have a narrow yield range, one with a balanced cost profile and the other with an ABC. The other two have a wide yield range and balanced and ABC cost profiles, respectively.

<u>Core Costs</u>

The first case tests the sensitivity of the solution cost to the cost of purchasing and disassembling cores (Figure 6-1). Demand was fixed at 50 units, as in the main experiment, and level of uncertainty was fixed at the medium level (st. dev. = 0.03). The cost per unit of purchasing and disassembling a core was varied from $10 to $115, where in the main experiment it was fixed at $80. Results were exactly as expected, with the optimal solution cost increasing almost linearly with core costs. When core costs approach $100, the solution cost flattens out. Recall that the expected value of a disassembled core was set at $100 across all problems in the problem set. When the cost of purchasing and disassembling a core is equal to the expected value of the first

disassembly, it no longer makes economic sense to disassemble, and a complete set of new parts is therefore purchased to meet demand. This is one way of crossing into the "manufacturing frontier" described in the results of the main experiment, since beyond this point the operation ceases to be remanufacturing and becomes manufacturing.

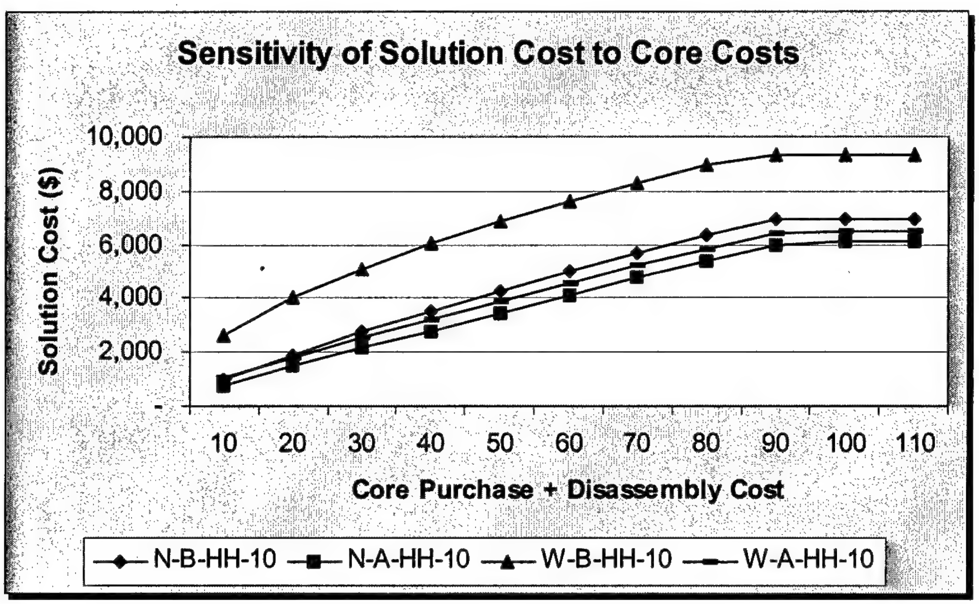


Figure 6-1: Sensitivity of Solution Cost to Core Costs

Figure 6-2 shows the optimal number of disassemblies Q^d across the same range of costs. The Q^d function appears as a step function of sorts, with discrete steps corresponding to the cost and yield structure of the products. For example, the products with balanced cost profiles (top two curves in Figure 6-2) show a series of distinct steps, associated with those points where the marginal value of disassembling an additional core drops relative to the increased cost. Conversely, the two products with an ABC cost profile have essentially two large steps. The first corresponds to the cost at which the "insignificant many" parts no longer benefit from increased disassemblies. The second

corresponds to the higher cost where the "significant few" parts, in this case two, no longer benefit. Between these two steps, for the ABC case, the optimal quantity of disassemblies remains constant. In all four cases, the optimal quantity of disassemblies drops to zero when (or before) the core cost equals the expected value of the core, as expected.

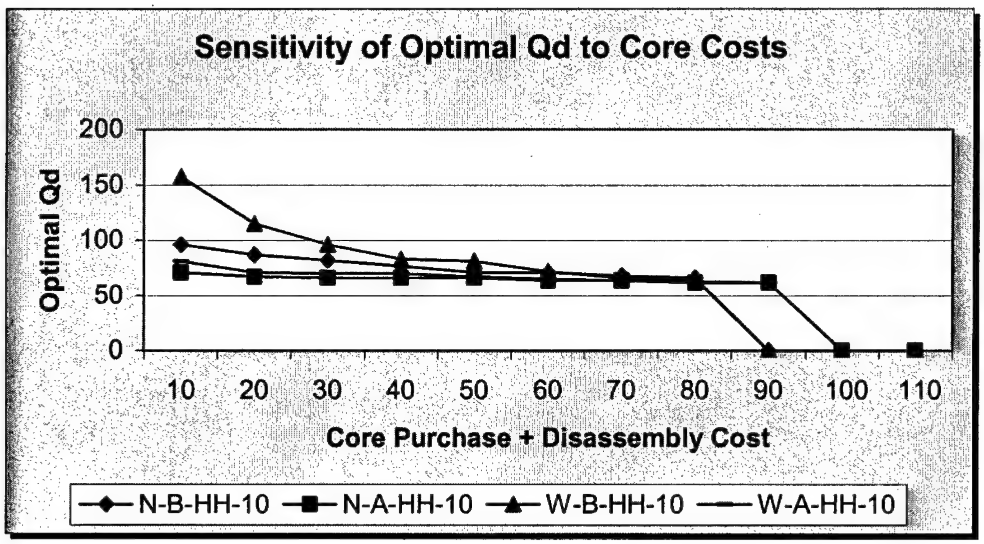


Figure 6-2: Sensitivity of Optimal Q^d to Core Costs

The final results (Figure 6-3) illustrate the sensitivity of the performance gap between ReNet and SIPR to core costs. When core purchase and disassembly costs are low and the product has an ABC cost profile, ReNet performs very poorly relative to SIPR. This result is consistent with expectations, since ReNet tends to disassemble fewer cores than SIPR as shown in the main experiment. When core costs are low, the lost opportunity of disassembling additional cores is greater in magnitude. Conversely, as core costs rise the advantage of disassembling additional cores is diminished and the solutions generated by the two techniques converge. When core costs are high, ReNet

fails to detect the point at which disassemblies cease to be economically attractive and its overall costs therefore rise sharply relative to those of SIPR for all product types.

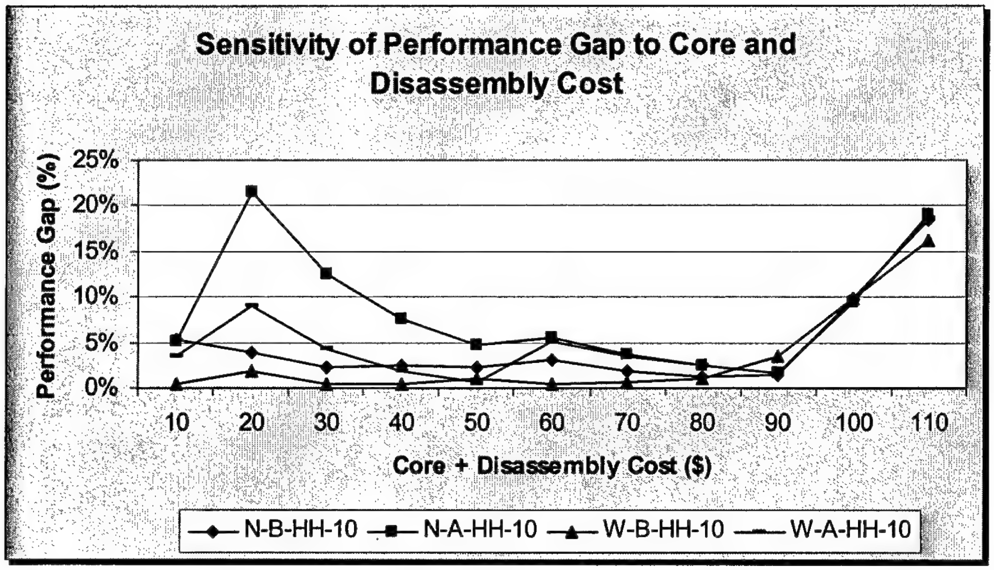


Figure 6-3: Sensitivity of Performance Gap to Core Costs

Target Service Level

The second sensitivity analysis tests the effect of varying the target service level, which was held constant at 0.95 in the main experiment (Figure 6-4). Demand was set at 50 units, and core costs at $80 as in the main experiment. Uncertainty remained at the medium (st. dev. = 0.03) level. The results are surprising, in that the solution cost is quite insensitive to the target service level. Common wisdom generally dictates a sharp rise in costs associated with service levels that approach 100%. The system service level has different characteristics from those of a single-part problem, however. At lower target service levels, the optimal quantity of disassemblies will supply most of the parts needed to reach the target. As the target increases, large gains in service level can be attained by

simply disassembling a few more cores and buying a few more parts. In fact, in many cases increasing a single part by one unit can increase the service level from 95% to over 99%.

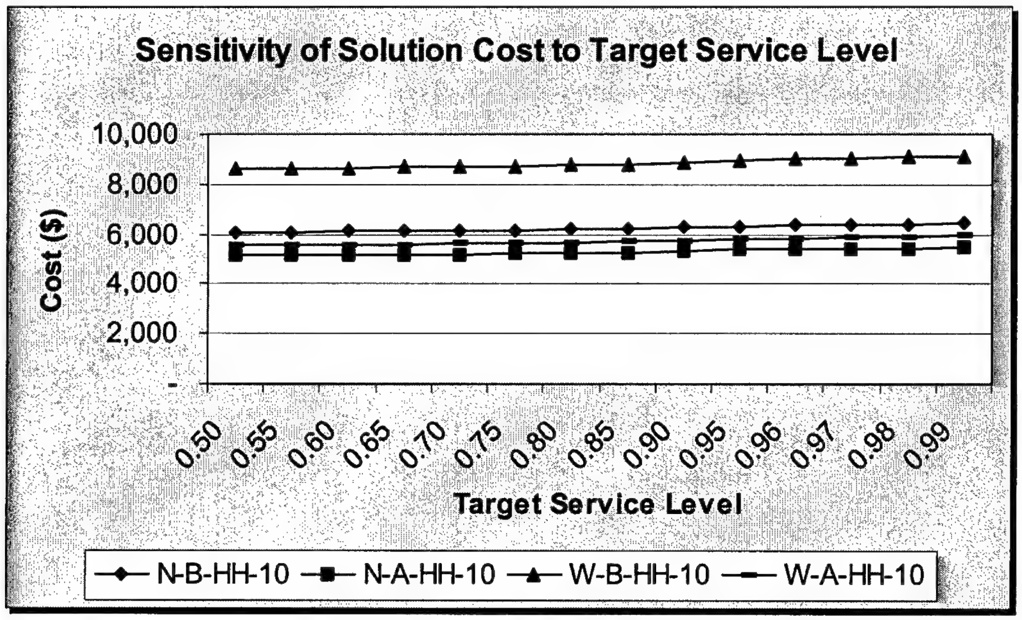


Figure 6-4: Sensitivity of Solution Cost to Target Service Level

There is a drawback to increased service level targets, however, and it comes in the form of excess parts. Figures 6-5 and 6-6 show the expected excess, in units and dollar value, respectively, for the same problems across the same range. Two points are noteworthy with respect to excess. First, the expected excess increases steadily with the target service level. And second, the excess oscillates, albeit around a steadily increasing mean. The former is consistent with expectations, while the latter is simply a by-product of the discrete structure of the problem.

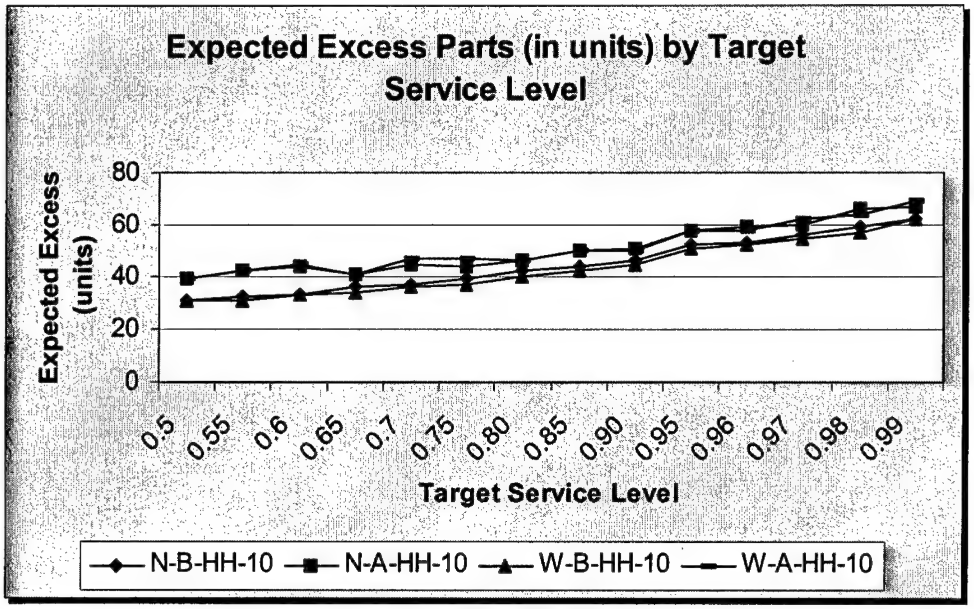


Figure 6-5: Expected Excess Parts(units) by Target Service Level

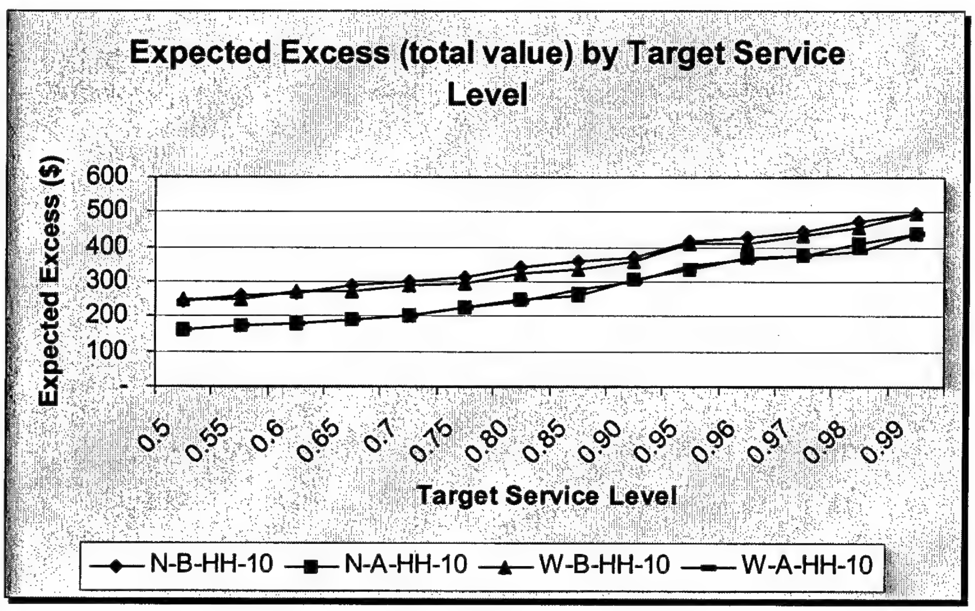


Figure 6-6: Expected Excess Parts (inventory value) by Target Service Level

End-Item Demand

The third factor tested is the end-item demand, and its associated effect on the cost and number of disassemblies. In this case, it should be noted that the cost reported is the per-unit cost, and likewise the number of disassemblies was transformed to a disassemblies-per-unit-demanded figure. Obviously, it makes little sense to compare aggregate solution costs when the baseline number of units is not equal across the data points. All other factors were held fixed as in the preceding experiments. Figure 6-7 shows the results of the unit cost across a range of demand from 1 to 100 units. It is clear that there is an economy of scale, but for these problems its effect diminishes beyond about a half-dozen units.

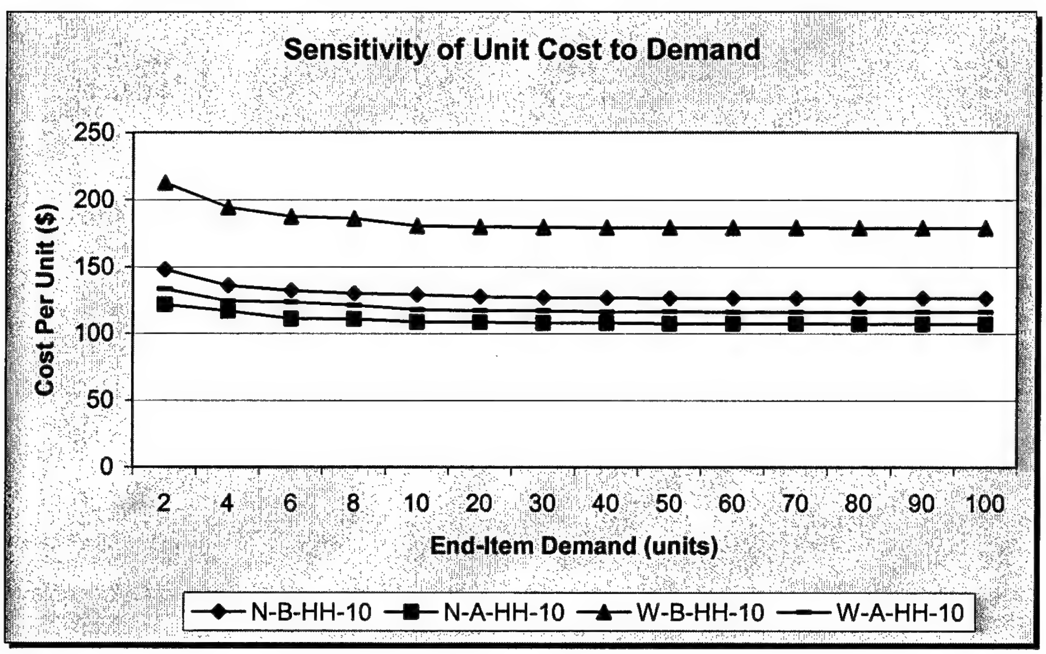


Figure 6-7: Sensitivity of Unit Cost to Unit Demand

Figure 6-8 shows the best available disassembly-to-assembly ratio across the same range. Again, the nature of the problem manifests itself as an oscillating discrete function that eventually converges, in this case at a demand of about 30 units. Beyond the point of convergence, the level of end-item demand has little effect on the optimal disassembly-to-assembly ratio.

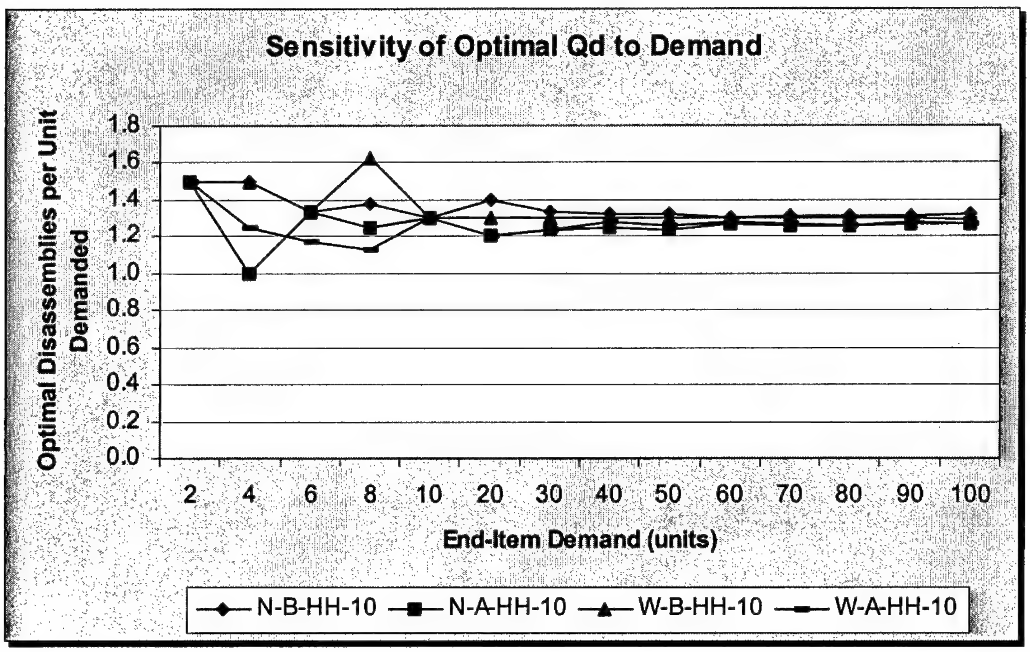


Figure 6-8: Sensitivity of Optimal Q^d to Demand

Figure 6-9 shows the performance gap between ReNet and SIPR across a range of demands. For lower demands, ReNet performs relatively poorly, although its performance oscillates in a discrete manner as before. For higher demands (about 20 units or more), ReNet's performance converges to within 3% of SIPR, while reaching a more steady-state.

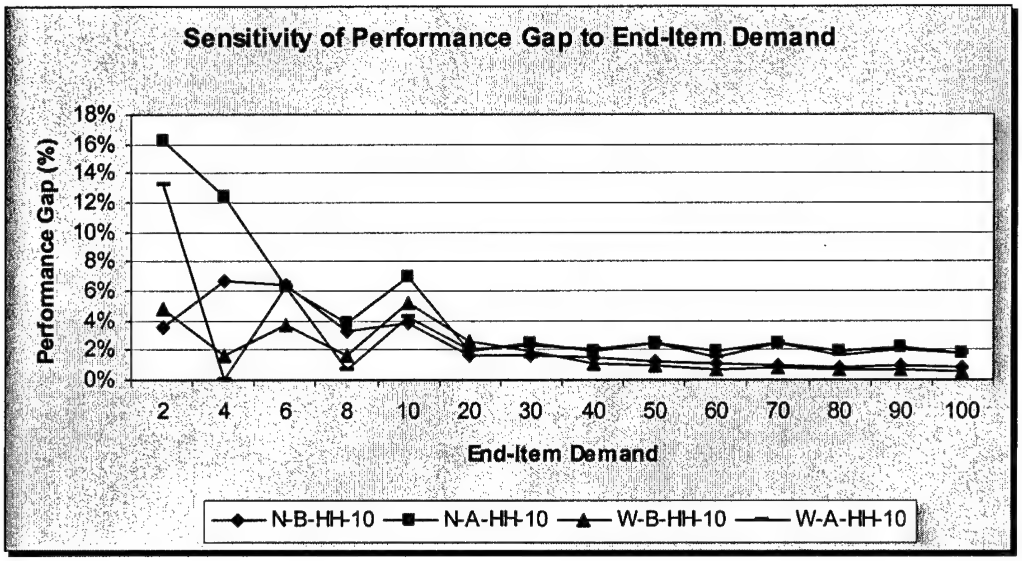


Figure 6-9: Sensitivity of Performance Gap to Demand

<u>Level of Uncertainty (Standard Deviation of Yield Distributions)</u>

The fourth and final sensitivity analysis was conducted on the uncertainty factor. As explained in chapter 5, the experimental levels of uncertainty in the main experiment were chosen so that the tails of the normal distribution could be contained within the feasible region of the problem, even for the highest-yield parts. To vary the uncertainty across a broader range, only products with a narrow yield range (0.35 to 0.65) could be used. The results that follow are from the two narrow-range products of the previous three tests, omitting the third and fourth.

Solution cost increases significantly with level of uncertainty, as expected (Figure 6-10). The SIPR cost of Problem 1 (Narrow, Balanced, HiHi) increased by 18.2% across the range tested, while the cost of Problem 2 (Narrow, ABC, HiHi) increased by 22.5%. Beyond the levels tested in the main experiment, the cost curve flattens, once again illustrating the "manufacturing frontier." The solution cost is higher for the balanced case across all levels, consistent with earlier results for the HiHi cost-yield match.

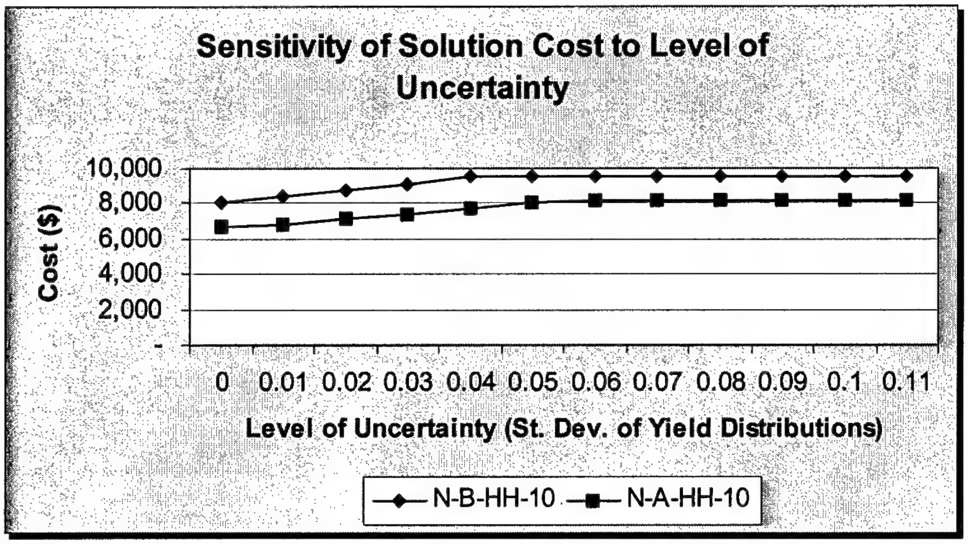


Figure 6-10: Sensitivity of Solution Cost to Level of Uncertainty

Figure 6-11 shows the optimal number of disassemblies as a function of uncertainty. The number increases steadily for increasing levels of uncertainty, which is consistent with the use of additional cores as safety stock, but each part has a unique level at which cores are no longer a viable option and the optimal quantity drops sharply to near-zero (the "manufacturing frontier" noted in the main experiment). The results notwithstanding, the highest level of uncertainty tested is more extreme than would normally be experienced. For example, the yield percentage of the part with an average yield of 0.65 would vary between 32% and 99% for the highest level. Still, the results make a convincing argument for reducing uncertainty of yields, either through design and engineering or through better estimation techniques like diagnostic testing.

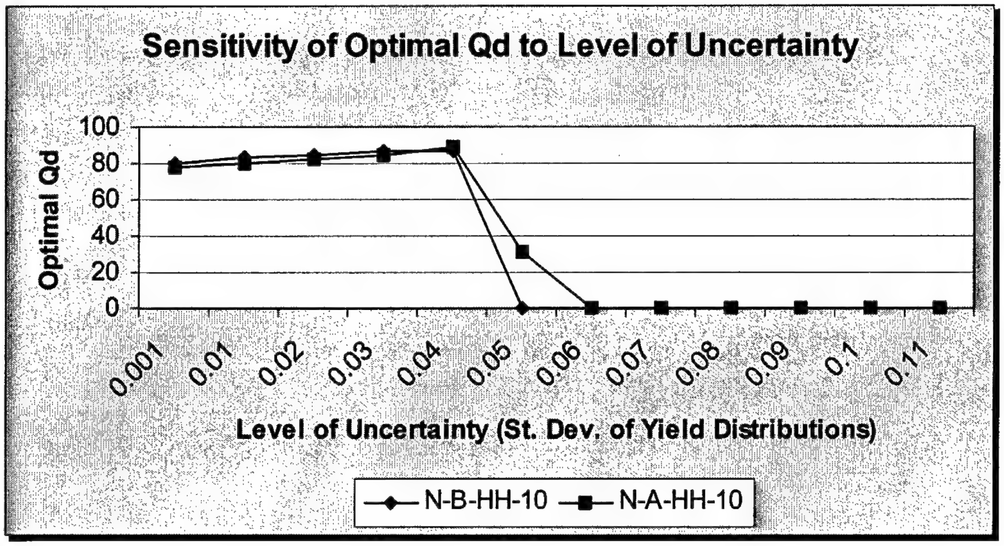


Figure 6-11: Sensitivity of Optimal Q^d to Level of Uncertainty

Although not included in the sensitivity experiment, HiLo products were shown in main experiment to reach the manufacturing frontier at a much lower level of uncertainty than shown here. This phenomenon occurred in all 8 HiLo cases, and in 6 of the 8 it occurred at the medium level of uncertainty (st. dev. = 0.03).

Finally, Figure 6-12 shows the effect of uncertainty on the performance gap between ReNet and SIPR. Earlier results indicated the level of uncertainty had a relatively small effect on the performance gap for HiLo problems. The results of the sensitivity test show clearly that the range of levels of uncertainty tested in the main experiment did not extend far enough to detect its effects on the performance gap. The results are consistent with those of the main experiment in the range tested (i.e. st. dev. = 0.01 to 0.05), but beyond that the gap widens steadily with increasing uncertainty. Once again, this is easily explained by the fact that ReNet fails to detect the threshold of the

manufacturing frontier, and therefore continues to disassemble cores even if it is not economical to do so.

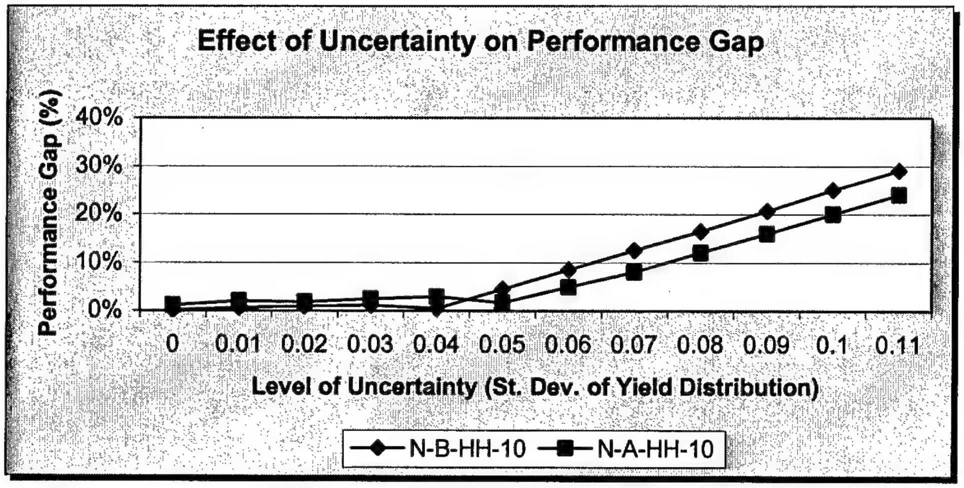


Figure 6-12: Effect of Uncertainty on Performance Gap

<u>Multi-Period Problem</u>

The results of the secondary experiment, which solved the four problems from the sensitivity analysis for a multi-period planning horizon, are now presented. SIPR uses the same solution methodology as before, but solves a single period at a time beginning with the most current and working forward through time. After solving each period, the expected excess associated with the solution is then carried over as starting inventory before solving the following period. Although it is capable of solving large, multi-period problems quickly in a deterministic sense, ReNet uses the same technique as SIPR. The reason for solving a series of single-period problems in the case of ReNet stems from the fact that, being a deterministic model, it does not account for the expected excess due to

stochastic yields. As with SIPR, the expected excess must be carried over and accounted for at the start of each period. Since it solves single-period problems in a few seconds, it still holds a large advantage in solution efficiency.

For the multi-period experiment, uncertainty was held fixed at the medium level of 0.03 as before. The number of periods was set at 5, and the total demand across the planning horizon was held constant at 350 units for all cases. The demand was varied in four ways to test any differences in the performance of the two techniques under a variety of patterns: Increasing, Decreasing, Spike, and Trough. The patterns are illustrated in Figure 6-13.

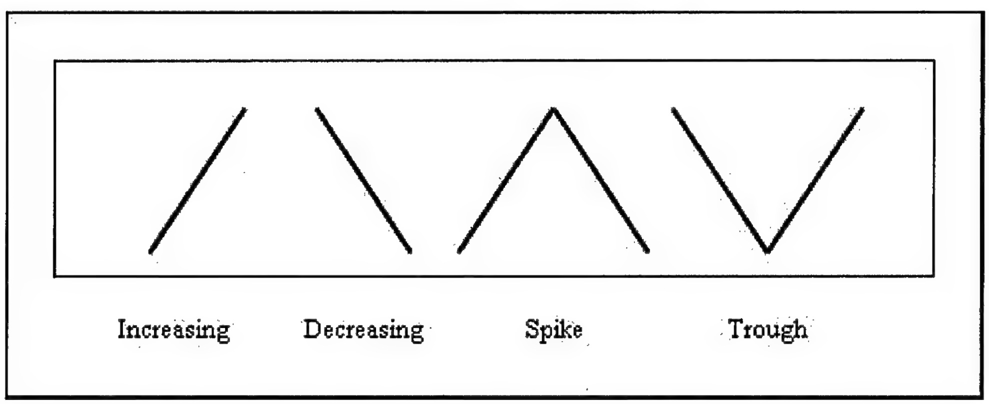


Figure 6-13: Demand Patterns used in Multi-Period Experiment

In addition, capacity was unconstrained as in the main experiment. Lead times were assumed to be zero to avoid confounding the results. And finally, order costs were assumed to be zero for the same reason.

Main Effects of Factors on Performance Gap

The main effects of the demand pattern on the performance gap between SIPR and ReNet are shown in Figure 6-14. Although differences are present, they are too small to draw any general conclusions about the main effect of demand pattern.

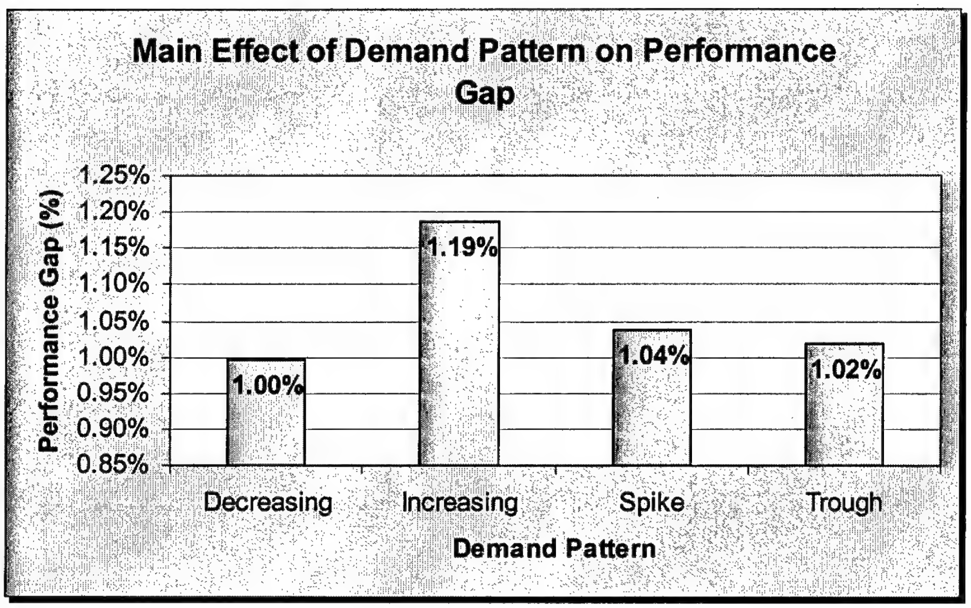


Figure 6-14: Main Effect of Demand Pattern on Performance Gap

The main effect of yield range on the performance gap remains consistent with earlier results for the same problems in the main experiment (Figure 6-15). Likewise, the higher gap for the ABC cost profile agrees with previous results (Figure 6-16). Note that the performance gap is actually smaller for the multi-period case than the corresponding single-period gaps. This result is consistent with expectations discussed in Chapter 3. Recall that SIPR is sub-optimal for the multi-period case, since the problem becomes intractable beyond a single-period. The results of the single-period case were therefore

expected to represent an upper bound on the performance gap, which is supported by the results reported here.

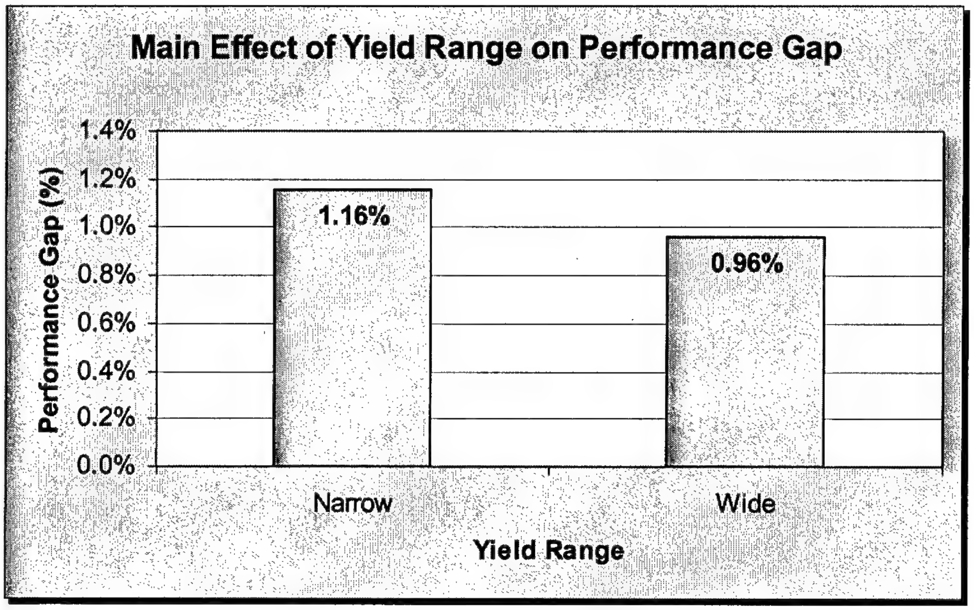


Figure 6-15: Main Effect of Yield Range on Performance Gap

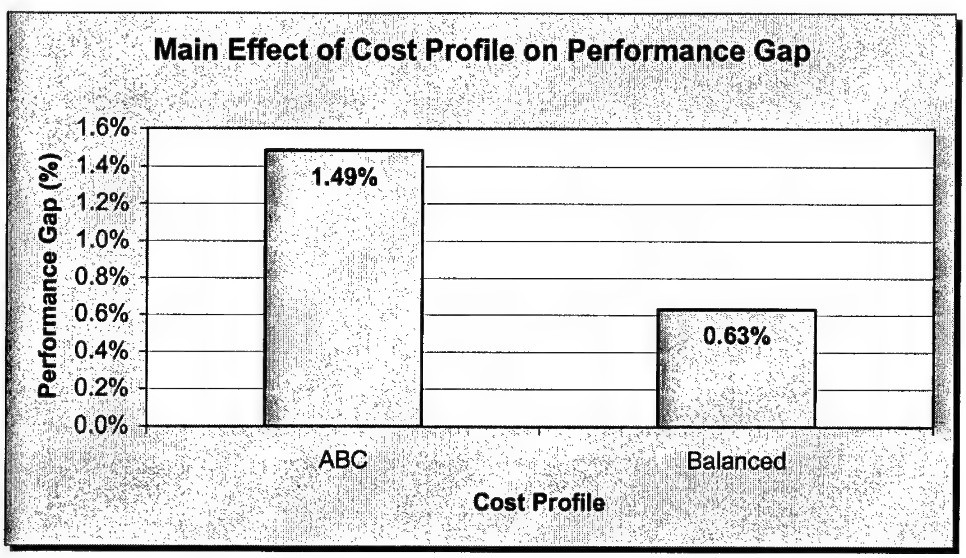


Figure 6-16: Main Effect of Cost Profile on Performance Gap

Main Effects of Factors on Solution Cost

The main effects on solution cost are now presented and compared with the corresponding single-period results with the exception of the demand pattern, which was not used in the main experiment. The solution costs for the following results are the SIPR solutions, although the claim of near-optimality no longer applies. As was the case in the performance gap comparison, the demand pattern has little effect on solution cost (Figure 6-17).

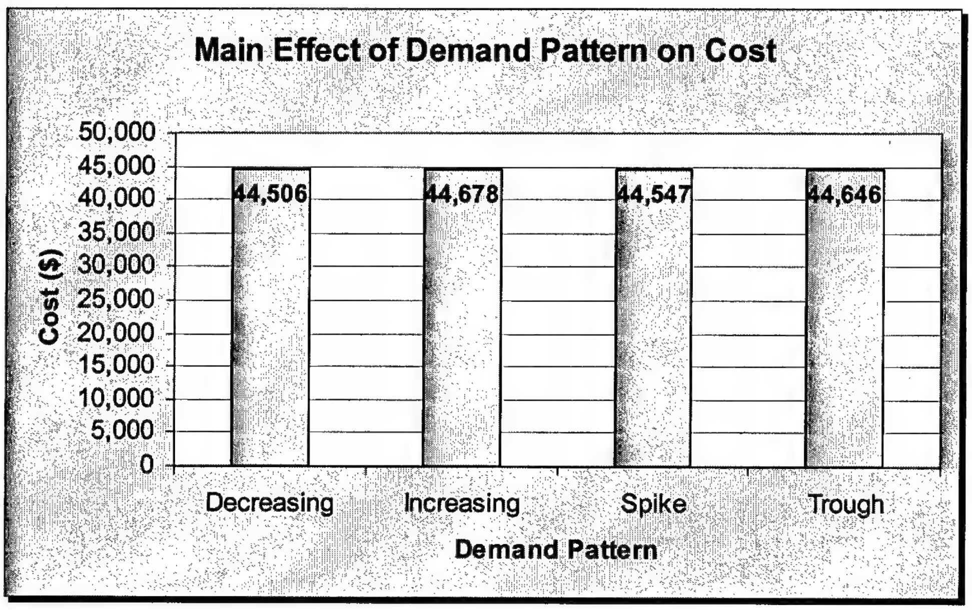


Figure 6-17: Main Effect of Demand Pattern on Cost

The main effect of yield range on the solution cost matches with the results found in the main experiment for the test problems (Figure 6-18), as does the main effect of cost profile (Figure 6-19).

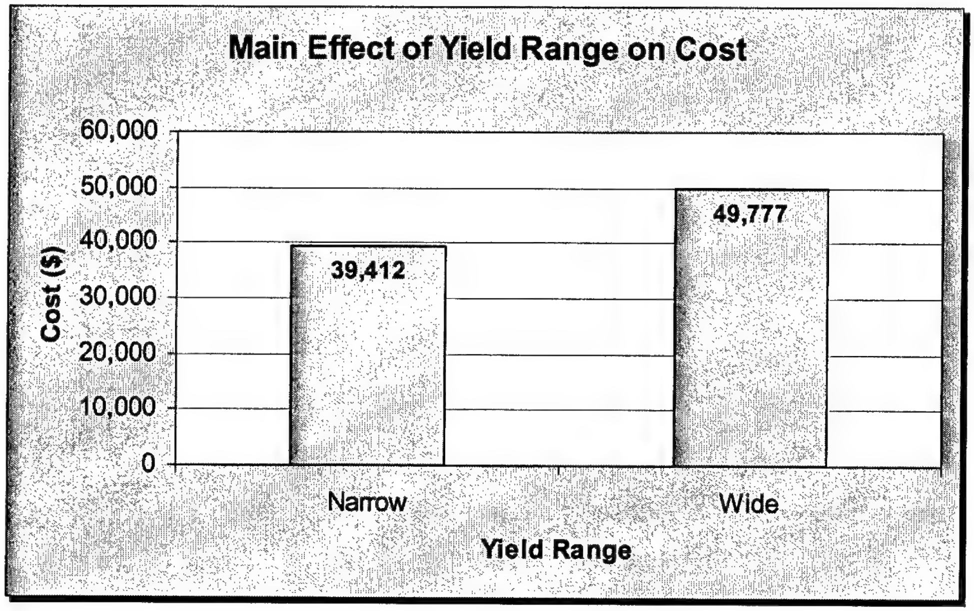


Figure 6-18: Main Effect of Yield Range on Cost

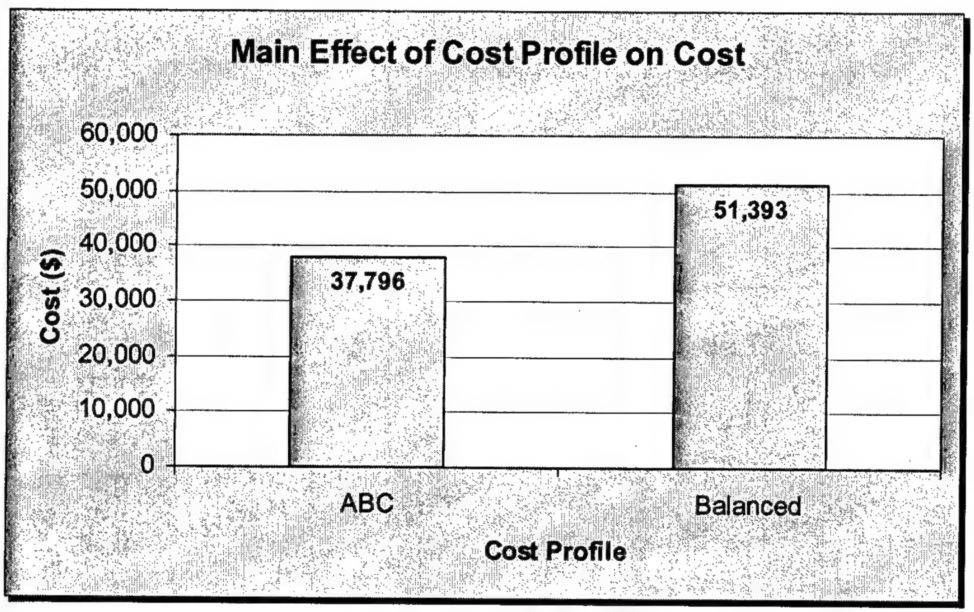


Figure 6-19: Main Effect of Cost Profile on Cost

Finally, the solution times are compared in Table 6-1 to show the difference in relative efficiency. Even for the multi-period case, ReNet solves in a few seconds. SIPR took just over 8 minutes on average, with a maximum of 9.6 minutes and a minimum of 7.3 minutes.

Table 6-1: Solution Times for Multi-period Problem

Technique	***Average Solution Time (seconds)****	***Maximum***	***Minimum***
SIPR	494.9	577.2	437.0
ReNet	1.5	1.7	1.4

*1.7 MHz processor with 256 MB random access memory (RAM)

Expected Excess at the End of the Planning Horizon

Due to the limited nature of the multi-period experiment (i.e. a single planning horizon), a final analysis of ending inventory is needed to determine the end-state of the planning horizon for each case. Recall that the ending inventory is charged a holding cost in the solution cost results reported to this point. But since the holding cost is low relative to disassembly and purchase costs, the inherent problem of increasing amounts of excess inventory building up over time is effectively masked. This section presents the end-states by technique, problem type, and demand pattern.

Figure 6-20 presents the ending inventory for each problem and for each of the two techniques. Two properties are suggested by the results. First, ReNet experiences slightly higher excess inventory levels for products with a balanced cost profile, while it experiences much lower excess levels for products with an ABC cost profile. This can be explained by the fact that SIPR disassembles a greater number of cores for ABC products, therefore accumulating more used parts for lower cost items. And second,

products with a wide range of yields accumulate less excess inventory than do those with a narrow range. This also stands to reason, since the overall yields of the lower-cost parts are much lower for this case, and so fewer parts will enter the system.

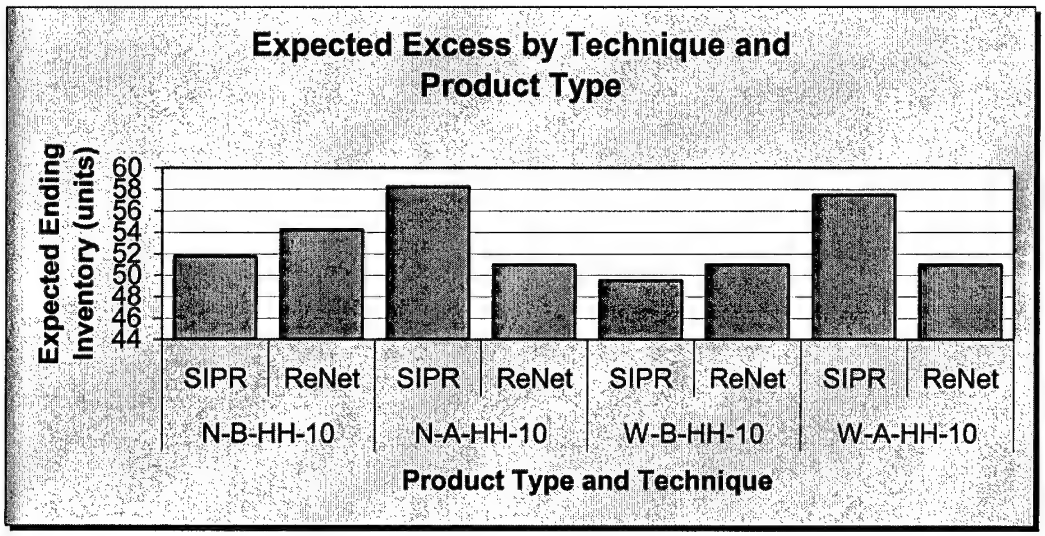


Figure 6-20: Expected Excess by Technique and Product Type

Averaging the excess across the two techniques, some general conclusions regarding demand patterns and product types can be drawn (Figures 6-21 and 6-22). First, it is clear that increasing demand throughout the planning horizon results in increased excess inventory (Figure 6-21). The increasing demand pattern showed the highest excess, since it is characterized by 5 periods of increasing demand, followed by the trough, which had 3 periods of increasing demand at the end of the horizon. In contrast, the decreasing and spike patterns showed significantly lower excess levels. Second, the ABC cost structure generates a greater number of excess parts than the balanced cost structure. And finally, a wide range of yields results in a lower number of excess parts than a narrow range, as shown previously in figure 6-20.

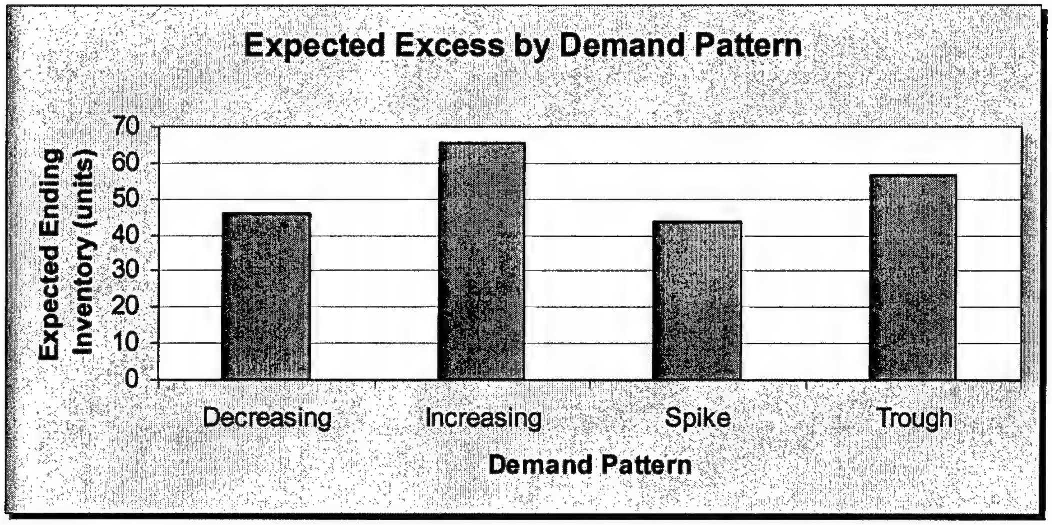


Figure 6-21: Expected Excess by Demand Pattern

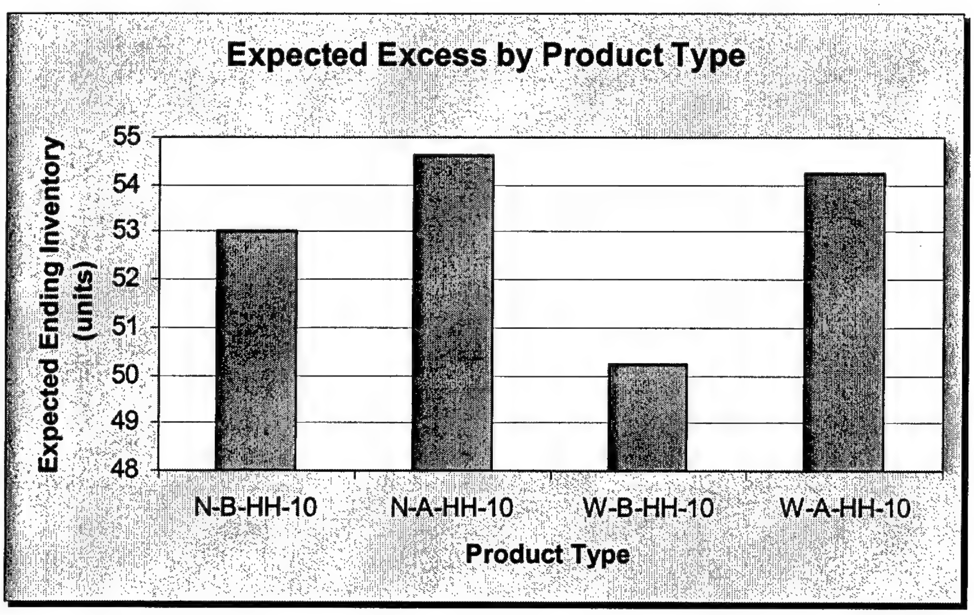


Figure 6-22: Expected Excess by Product Type

Summary of Multi-Period Results

In total, the results of the multi-period case were consistent with those of the single-period case. The performance gap for each was about 1.5% on average for the four problems tested in the secondary experiment, for example.

The one surprising result was the relatively slow rate at which excess parts accumulate. It was initially thought that the high-yield parts would carry excess that would never be used, since the condition would repeat for each period. The results of the secondary experiment temper that expectation, in that both methods continually adjusted their solutions to effectively use the excess parts in each period. Although the excess parts continually increase, they do so at a slower rate than expected.

Conclusions

The results of the set of four sensitivity experiments largely confirm the results of the main experiment. They also suggest that the performance gap between ReNet and SIPR will widen beyond the levels of uncertainty tested in the main experiment. Finally, the sensitivity analysis on the level of uncertainty helped to confirm the existence of a "manufacturing frontier," which was noted in the main experiment for the HiLo problems.

A final secondary experiment was also reported, in which multi-period heuristic versions of ReNet and SIPR were tested for a 5-period problem. The results were generally consistent with those of the single-period case. Both techniques performed well, and surprisingly made efficient use of excess inventory from period to period. In fact the level of inventory, while oscillating, remained relatively level.

Chapter 7 concludes the dissertation, offering more discussion on the implications of the research results to remanufacturing firms. It also offers suggestions for further research.

CHAPTER 7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

The preceding two chapters presented the detailed results of the main and secondary experiments and the sensitivity analyses. They also offered implications of the results in each section. This chapter summarizes the results and implications, and offers suggestions for future research.

The most compelling result from the research was the dramatic effect of the cost-yield match factor, and its strong interactions with the other factors in the study. Throughout the results, it became consistently clear that products with high cost, low yield parts are very expensive to remanufacture. In general, it becomes economically unattractive to remanufacture such products even for relatively low levels of yield uncertainty. The implication for remanufacturing firms is clear: they should focus resources on the design of high-cost parts to improve their reliability, or cease remanufacturing. The strong interaction between the cost-yield match and the other factors underscores this result in that their effects are exaggerated, in some cases significantly, for those products with high-cost, low-yield parts.

Another recurring theme throughout the results is the existence of a "manufacturing frontier," where remanufacturing becomes more expensive than manufacturing new products. Clearly this is an important strategic concept in the industry. The research illuminated several factors that determine the location of the

frontier. The cost-yield match factor was the most significant determinant. High-cost, low-yield parts drive costs up significantly in all cases, pinching the cost gap between manufacturing and remanufacturing and making the latter a less attractive option. High yield uncertainty has a similar effect, which is exacerbated by the HiLo cost-yield match. And to a lesser degree, a wide yield range and balanced cost structure increase costs for HiHi products, making remanufacturing less economically viable.

The results of the first research question indicate that SIPR outperforms ReNet to varying degrees, but on average ReNet performs well. However the sensitivity analyses illustrated that ReNet is less robust, particularly in regard to detecting and adjusting for the manufacturing frontier property. So although it performed well on average, ReNet is a riskier technique, especially for products with a HiLo cost-yield match and a high degree of yield uncertainty.

Research question 2 confirmed expectations that a higher level of yield uncertainty will increase costs, although it has no effect beyond the manufacturing frontier. In general, higher yield uncertainty leads to fewer disassemblies and a greater number of new parts, driving up costs. Product complexity, as modeled by the number of parts, exaggerates the effect of uncertainty on parts. The implication for remanufacturing firms is that a reduction in yield uncertainty, particularly for high-cost parts, will result in significant savings. This reduction is possible in all phases of a product life cycle, beginning with design and continuing through quality control to actual monitoring and diagnosis of usage and failures.

The results of the third and final research question largely underscored the results shown elsewhere. The cost-yield match has the greatest overall impact on cost, while the

cost profile and yield range also have significant effects that interact with the match. The general rules of thumb, summarized in Table 5-14, indicate that a HiHi match, narrow yield range, and ABC cost profile are the most desirable attributes in terms of cost reduction. While the latter will generally be beyond the firm's scope of control, the first two can be manipulated given attention and/or capital investment.

Suggestions for Future Research

Remanufacturing is already a large activity in the U.S., and indications point to increased growth in the future. Still, it has remained largely hidden and has received little attention in the literature. This research begins to open the door for more research in the area of remanufacturing inventory planning.

Perhaps the greatest research opportunity stemming from this research is an increased focus on the multi-period case. With more and more small- to mid-sized companies implementing ERP systems and using their MRP modules for inventory planning, the techniques suggested here can be improved to provide firms the capability to calculate requirements across a planning horizon within an MRP framework. As discussed in Chapter 3, the complexity of the problem grows quickly with the number of periods, so the natural extension of this research lies in the development of efficient heuristics that address the stochastic properties of the problem.

Another potential extension lies in the simplifying assumptions of deterministic end-item demand and unconstrained core supply and lead time. Explicit modeling of these additional stochastic factors would increase the robustness of the models and allow

firms to more accurately plan requirements. The focus on yield uncertainty presented here was a first step toward that end.

Finally, this research used an experimental set of simplified problems to show the behavior of the problem clearly without confounding effects. Although these problems were rooted in real cases and were designed to capture the extreme values of the experimental factors, future research using empirical tests with actual firm data would help to validate the results. Such research would also likely suggest improvements to the models.

APPENDIX A – Problem Set

Problem	Yield Range	Cost Profile	Cost-Yield Match	Number of Parts	Part #	Avg. Yield	New Cost
N-B-HH-10	Narrow	Balanced	HiHi	10	1	0.850	18.00
					2	0.820	17.46
					3	0.790	17.00
					4	0.760	15.00
					5	0.720	15.00
					6	0.680	14.00
					7	0.640	13.00
					8	0.610	11.00
					9	0.580	10.00
					10	0.550	8.00
					Manufacturing Cost		138.46
N-B-HL-10	Narrow	Balanced	HiLo	10	1	0.450	28.00
					2	0.420	31.00
					3	0.390	32.00
					4	0.360	33.00
					5	0.320	34.00
					6	0.280	35.00
					7	0.240	37.00
					8	0.210	37.00
					9	0.180	38.00
					10	0.150	39.00
					Manufacturing Cost		344.00
N-A-HH-10	Narrow	ABC	HiHi	10	1	0.850	56.00
					2	0.820	42.00
					3	0.790	8.00
					4	0.760	6.00
					5	0.720	7.00
					6	0.680	1.00
					7	0.640	0.70
					8	0.610	0.60
					9	0.580	0.50
					10	0.550	0.47
					Manufacturing Cost		122.27

Problem	Yield Range	Cost Profile	Cost-Yield Match	Number of Parts	Part #	Avg. Yield	New Cost
N-A-HL-10	Narrow	ABC	HiLo	10	1	0.450	0.80
					2	0.420	1.00
					3	0.390	1.20
					4	0.360	8.64
					5	0.320	20.00
					6	0.280	20.00
					7	0.240	25.00
					8	0.210	30.00
					9	0.180	213.00
					10	0.150	220.00
					Manufacturing Cost		539.64
W-B-HH-10	Wide	Balanced	HiHi	10	1	0.850	24.00
					2	0.770	22.39
					3	0.690	21.00
					4	0.610	20.00
					5	0.530	19.00
					6	0.470	18.00
					7	0.390	17.00
					8	0.310	16.00
					9	0.230	15.00
					10	0.150	14.00
					Manufacturing Cost		186.39
W-B-HL-10	Wide	Balanced	HiLo	10	1	0.850	16.00
					2	0.770	17.61
					3	0.690	19.00
					4	0.610	20.00
					5	0.530	21.00
					6	0.470	22.00
					7	0.390	23.00
					8	0.310	24.00
					9	0.230	25.00
					10	0.150	26.00
					Manufacturing Cost		213.61
W-A-HH-10	Wide	ABC	HiHi	10	1	0.850	58.00
					2	0.770	46.84
					3	0.690	8.00
					4	0.610	6.00
					5	0.530	5.00
					6	0.470	4.50
					7	0.390	1.00
					8	0.310	0.80
					9	0.230	0.14
					10	0.150	0.10
					Manufacturing Cost		130.38

Problem	Yield Range	Cost Profile	Cost-Yield Match	Number of Parts	Part #	Avg. Yield	New Cost
W-A-HL-10	Wide	ABC	HiLo	10	1	0.850	1.00
					2	0.770	1.50
					3	0.690	2.00
					4	0.610	4.00
					5	0.530	16.00
					6	0.470	18.00
					7	0.390	20.00
					8	0.310	21.00
					9	0.230	162.70
					10	0.150	170.00
					Manufacturing Cost		416.20
N-B-HH-5	Narrow	Balanced	HiHi	5	1	0.850	32.00
					2	0.775	30.00
					3	0.700	27.04
					4	0.625	27.00
					5	0.550	25.00
					Manufacturing Cost		141.04
N-B-HL-5	Narrow	Balanced	HiLo	5	1	0.450	64.00
					2	0.375	65.67
					3	0.300	66.00
					4	0.225	71.00
					5	0.150	72.00
					Manufacturing Cost		338.67
N-A-HH-5	Narrow	ABC	HiHi	5	1	0.850	97.00
					2	0.775	11.00
					3	0.700	10.95
					4	0.625	1.30
					5	0.550	1.00
					Manufacturing Cost		121.25
N-A-HL-5	Narrow	ABC	HiLo	5	1	0.450	5.00
					2	0.375	5.00
					3	0.300	48.08
					4	0.225	50.00
					5	0.150	468.00
					Manufacturing Cost		576.08
W-B-HH-5	Wide	Balanced	HiHi	5	1	0.850	44.00
					2	0.675	40.40
					3	0.500	38.00
					4	0.325	35.00
					5	0.150	33.00
					Manufacturing Cost		190.40

Problem	Yield Range	Cost Profile	Cost-Yield Match	Number of Parts	Part #	Avg. Yield	New Cost
W-B-HL-5	Wide	Balanced	HiLo	5	1	0.850	35.00
					2	0.675	38.00
					3	0.500	43.00
					4	0.325	48.00
					5	0.150	50.00
					Manufacturing Cost		214.00
W-A-HH-5	Wide	ABC	HiHi	5	1	0.850	101.00
					2	0.675	12.50
					3	0.500	9.82
					4	0.325	2.00
					5	0.150	1.00
					Manufacturing Cost		126.32
W-A-HL-5	Wide	ABC	HiLo	5	1	0.850	4.00
					2	0.675	7.00
					3	0.500	37.00
					4	0.325	50.00
					5	0.150	380.80
					Manufacturing Cost		478.80

APPENDIX B – ReNet Visual Basic® Code

The ReNet technique is comprised of three Visual Basic® (VBA) programs and the ILOG CPLEX optimizer. The first VBA program builds the problem file in standard LP format from the user input spreadsheet. The problem file is then imported into CPLEX for solution of the deterministic problem. The second VBA program imports and formats the CPLEX solution. The third program adds the safety quantities to the deterministic solution.

1. Problem File Generator

The first program creates a text file for the problem as defined by the user in a spreadsheet. In the spreadsheet, the structure of the network is first defined in terms of the number and types of nodes and arcs. On a separate worksheet, the specific problem characteristics (demands, costs, etc.) are entered. The result is a text file representing the problem in standard LP format, which can be imported and solved by an LP optimization algorithm like CPLEX.

Variable declaration

```
Dim i As Integer, j As Integer, k As Integer
Dim Node1 As Integer, Node2 As Integer, Part As Integer, Period As Integer
Dim NumNodes As Integer, NumParts As Integer, NumPeriods As Integer
Dim NumCores As Integer
Dim Filename As String, Pathfile As String, OutputLog As String
Dim Core As Integer, QPA As String
Dim HoldingCostPct As Double, WeeklyHC As Double, WTIPrice As Double
Dim DisassemblyCost As Double
```

```
Dim Profit As Double, Cost As Double
Dim ScrapPct As Double, RemanPct As Double, CapacityLimit As Integer
Dim Demand As Integer
Dim Omit As Boolean, Done As Boolean
Dim InArray() As Integer, OutArray() As Integer, PartsArray() As String
Dim DemandArray() As Integer, CoreDemand As Integer, PartDemand As Integer
Dim Arc As String, Arc2 As String, NodeMatrix() As Integer
Dim CoreDemandArray() As Integer
Dim EveryFifth As Integer, RemanPrices() As Double
Dim Constraint As Long, CoreCost As Double, NodeType As String
Dim FlowType As String
Dim SupplyArray() As Integer, Destination As String, BegInv As Integer
Dim ArcCounter As Long, ArcArray() As String, ScrapOut As String
Dim RemanNode As Integer, CoreBuys As String, SellUsed As String
Dim PartsD As String, UnitsD As String, DemandConstraint As String
Dim ReceiptArray() As Integer, IntegerSolution As String
Dim ProblemNum As Integer, NumProbs As Integer, Anchor1 As Integer
Dim Anchor2 As Integer
```

The "Main" sub initializes the variables, reads in the problem data, and calls the other subroutines to create the problem file. First the variables are initialized and the number of problems is determined.

```
Public Sub Main()

i = 0
j = 0
k = 0
ArcCounter = 0
Constraint = 0

With Worksheets("Data").Range("A2")
  For i = 1 To 500
    If .Offset(i, 0) <> "" Then
      NumProbs = .Offset(i, 0)
    End If
  Next
End With
```

The anchor variables are initialized, and are later used to mark the upper leftmost cell of the current problem.

```
Anchor1 = 0
Anchor2 = 0
```

Now begin to create the problem files one at a time.

```
For ProblemNum = 1 To NumProbs
  Anchor1 = Anchor2
  With Worksheets("Data").Range("A2")
    j = 0
    Do
      j = j + 1
      If .Offset(Anchor1 + j, 4) <> "" Then
        NumParts = .Offset(Anchor1 + j, 4)
      End If
    Loop Until .Offset(Anchor1 + j, 4) = ""
    Anchor2 = Anchor1 + j
  End With
```

First read in the problem parameters

```
  With Worksheets("File Generator").Range("A1")
    .Offset(0, 1) = ProblemNum
    Filename = .Offset(0, 1).Value
    .Offset(1, 1) = NumParts
    NumPeriods = .Offset(2, 1)
    NumCores = .Offset(3, 1)
    CapacityLimit = .Offset(4, 1)
    DisassemblyCost = .Offset(5, 1)
    HoldingCostPct = .Offset(6, 1) / (100 * 52)
    CoreCost = .Offset(7, 1)
    UnitsD = .Offset(0, 3)
    PartsD = .Offset(1, 3)
    CoreBuys = .Offset(2, 3)
    SellUsed = .Offset(3, 3)
    IntegerSolution = .Offset(4, 3)
    OutputLog = .Offset(5, 3)
  End With

  With Worksheets("Balanced Nodes").Range("A1")
    NumNodes = .Offset(0, 1)
    RemanNode = .Offset(1, 1)
  End With
```

Next redimension and fill the arrays to store the network definition parameters

```
  ReDim InArray(NumNodes)
  ReDim OutArray(NumNodes)
  ReDim NodeMatrix(NumNodes, NumNodes)
  ReDim PartsArray(NumParts, 6)
```

```
    ReDim DemandArray(NumCores + 1, NumParts, NumPeriods)
    ReDim CoreDemandArray(NumCores, NumPeriods)
    ReDim CorePrices(NumCores)
    ReDim SupplyArray(NumCores)
    ReDim RemanPrices(NumCores)
    ReDim ReceiptArray(NumCores, NumPeriods)

    Call FillNodeMatrix
    Call FillPartsArray
    Call FillCoreDemand
    Call FillDemandArray
    Call FillRemanPrices
    Call FillSupplyArray
    Call FillReceiptArray
```

Open and build the output file

```
    Pathfile = "C:\D\I-M\" & Filename & ".lp"
    Open Pathfile For Output As #1
    Pathfile = "C:\D\P-M\" & OutputLog & ".txt"
    Open Pathfile For Output As #2

    Print #1, "Problem Parameters"
    Call WriteParameters
    Print #1, "MINIMIZE"
    Call ObjectiveFunction
    Print #1, "SUBJECT TO"
    Call Balance          'enforces balance for all nodes
    Call DemandNodes      'enforces that demands are met at demand nodes
    Call SupplyNodes      'ensures beginning supply is entered
    Call Reman            'ensures appropriate reman percentage
    Call Capacity         'enforces disassembly capacity per period
    Call Scrap            'ensures appropriate scrap percentage
    Call Multipliers      'ensures correct number of each part emanate from disassembly
    'Print #1, "BOUNDS"
    'Call NonNegativity   'enforces non-negativity for all variables
    If IntegerSolution = "Y" Then
      Print #1, "GENERAL"
      Call Integers       'ensures all flows are integral...optional
    End If

    Print #1, "End";
    Close #1
    Close #2
Next
```

```
End Sub
```

The following sub saves the problem parameters to a text file for reference

```
Sub WriteParameters()

With Worksheets("File Generator").Range("A1")
  Write #2, "PROBLEM PARAMETERS:"
    For i = 0 To 7
      Write #2, .Offset(i, 0) & " = ";
      Write #2, .Offset(i, 1)
    Next
    For i = 0 To 6
      Write #2, .Offset(i, 2) & " = ";
      Write #2, .Offset(i, 3)
    Next
  Write #2, ""
End With

With Worksheets("Balanced Nodes").Range("W3")
  Write #2, "NODE CHARACTERISTICS:"
    For i = 0 To 6
      For j = 0 To 3
        Write #2, .Offset(i, j) & " ";
      Next
      Write #2, ""
    Next
End With

With Worksheets("Balanced Nodes").Range("W17")
  Write #2, "INTEGER CONSTRAINTS:"
  If IntegerSolution = "N" Then
    Write #2, "No Integer Constraints Specified"
  ElseIf IntegerSolution = "Y" Then
    For i = 1 To NumNodes
      If .Offset(i, 0) <> "" Then
        Write #2, .Offset(i, 0);
        Write #2, .Offset(i, 1);
        Write #2, .Offset(i, 2)
      End If
    Next
  End If
  Write #2, ""
End With

Write #2, "NODE MATRIX:"
```

```
For Node1 = 1 To NumNodes
   For Node2 = 1 To NumNodes
      Write #2, NodeMatrix(Node1, Node2);
   Next
   Write #2, ""
Next
Write #2, ""

Write #2, "REMANUFACTURED UNIT PRICES:"
For Core = 1 To NumCores
   Write #2, "Core " & Core & " = " & RemanPrices(Core) & " "
Next
Write #2, ""
Write #2, ""

Write #2, "CORE DEMANDS:"
For Core = 1 To NumCores
   Write #2, "Core " & Core & ": ";
   For Period = 1 To NumPeriods
      Write #2, CoreDemandArray(Core, Period);
   Next
   Write #2, ""
Next
Write #2, ""

Write #2, "PROJECTED CORE RECEIPTS:"
For Core = 1 To NumCores
   Write #2, "Core " & Core & ": ";
   For Period = 1 To NumPeriods
      Write #2, ReceiptArray(Core, Period);
   Next
   Write #2, ""
Next

End Sub
```

The following sub builds the objective function and writes it to the text file

```
Sub ObjectiveFunction()

'First the direct costs within each period...these are activity costs
'First we may need to buy cores

If CoreBuys = "Y" Then
   For Period = 1 To NumPeriods
      For Core = 1 To NumCores
```

```
      Node1 = 1
      NodeType = "S"
      Call CoreBuyArc
      Print #1, " + " & CoreCost & Arc
    Next
  Next
End If


'Now the cost of disassembling all cores in all periods

  For Period = 1 To NumPeriods
    For Core = 1 To NumCores
      Call DisassemblyArc
      If Core <> NumCores Then
        Print #1, " + " & DisassemblyCost & Arc;
      Else
        Print #1, " + " & DisassemblyCost & Arc
      End If
    Next
  Next


'Now comes the cost of remanufacturing used parts

With Worksheets("Data").Range("A2")

  For Period = 1 To NumPeriods
  i = 0
    For Part = 1 To NumParts
      If .Offset(Part, 11) = "Y" Then
        i = i + 1
        EveryFifth = i Mod 5
        Cost = .Offset(Part, 14)
        Call RemanArc
        If Part < NumParts And EveryFifth > 0 Then 'This formats the .txt file
          Print #1, " + " & Cost & Arc;
        Else
          Print #1, " + " & Cost & Arc
        End If
      End If
    Next
  Next
```

```
'Finally, we need to buy new parts for final assembly

If Worksheets("Balanced Nodes").Range("A3").Offset(5, 22) <> "" Then

  For Period = 1 To NumPeriods

    For Part = 1 To NumParts

      Cost = PartsArray(Part, 1)
      Call NewPartArc
      If Part < NumParts And Part Mod 5 > 0 Then
        Print #1, " + " & Cost & Arc;
      ElseIf Part = NumParts Or Part Mod 5 = 0 Then
        Print #1, " + " & Cost & Arc
      End If
    Next
  Next

End If


'Now we need to add the holding costs from period to period

  For Node1 = 1 To NumNodes
    If NodeMatrix(Node1, Node1) > 0 Then

      For Period = 1 To NumPeriods

        If Node1 = 1 Then
          Cost = Round(CoreCost * (HoldingCostPct / 52), 6)
              'holding cost above does not include hurdle rate, since there
              'is no cost for cores...only warehousing, insurance, etc.
          For Core = 1 To NumCores
            Call HoldingCostArc
            If Core <> NumCores And Core Mod 5 > 0 Then
              Print #1, " + " & Cost & Arc;
            ElseIf Core = NumCores Or Core Mod 5 = 0 Then
              Print #1, " + " & Cost & Arc
            End If
          Next
        ElseIf Node1 = 4 Then
          For Part = 1 To NumParts
            Cost = Round((HoldingCostPct / 52) * PartsArray(Part, 2), 6)
            Call HoldingCostArc
```

```
                If Part <> NumParts And Part Mod 5 > 0 Then
                    Print #1, " + " & Cost & Arc;
                ElseIf Part = NumParts Or Part Mod 5 = 0 Then
                    Print #1, " + " & Cost & Arc
                End If
            Next
        ElseIf Node1 = 5 Then
            For Part = 1 To NumParts
                Cost = Round((HoldingCostPct / 52) * PartsArray(Part, 1), 6)
                Call HoldingCostArc
                If Part <> NumParts And Part Mod 5 > 0 Then
                    Print #1, " + " & Cost & Arc;
                ElseIf Part = NumParts Or Part Mod 5 = 0 Then
                    Print #1, " + " & Cost & Arc
                End If
            Next

        End If

      Next
    End If
  Next
End With

'And now the good stuff...the profits from sales
'First the profits from selling remanufactured and experienced parts

With Worksheets("Data").Range("A2")
For Node1 = 1 To NumNodes
  NodeType = Worksheets("Balanced Nodes").Range("A3").Offset(Node1, 23)
  If NodeType = "DP" Or NodeType = "DB" Then
    For Part = 1 To NumParts
      If .Offset(Part, 11) = "Y" Then
        For Period = 1 To NumPeriods
            Profit = .Offset(Part, 15)
            Call PartProfitArc
            If Period <> NumPeriods Then
              Print #1, " + " & Profit & Arc;
            Else
              Print #1, " + " & Profit & Arc
            End If
        Next
      End If
      If SellUsed = "Y" Then
        If .Offset(Part, 12) = "Y" Then
          For Period = 1 To NumPeriods
```

```
                Profit = .Offset(Part, 15)
                Call PartProfitArc

                If Period <> NumPeriods Then
                   Print #1, " + " & Profit & Arc;
                Else
                   Print #1, " + " & Profit & Arc
                End If
             Next
          End If
       End If
    Next
  End If
Next
```

```
'Now the profit from the sale of remanufactured units, calculated by applying the entire
'unit profit to part 1, which is used on all three assembly numbers and has a QPA of 1,
'meaning the number used will equal the number of units sold

  For Node1 = 1 To NumNodes
     If Worksheets("Balanced Nodes").Range("A3").Offset(Node1, 22) <> "" Then
        NodeType = Worksheets("Balanced Nodes").Range("A3").Offset(Node1, 23)
        If NodeType = "SDU" Then
           Profit = RemanPrices(1)
           For Period = 1 To NumPeriods
              Call UnitProfitArc
              If Period = NumPeriods Then
                 Print #1, " + " & Arc
              Else
                 Print #1, " + " & Arc;
              End If
           Next
        End If
     End If
  Next

End With

End Sub
```

The remaining subs build the constraints associated with nodes and arcs.

The sub below creates balance constraints for all nodes. This is a standard network LP constraint set.

```
Sub Balance()

With Worksheets("Balanced Nodes").Range("A3")

'The following nested loops build balance constraints for each node by part and period,
'respectively.

  For Node1 = 1 To NumNodes

'First, we get the specifics on the node
    NodeType = .Offset(Node1, 23)
    FlowType = .Offset(Node1, 24)
    ScrapOut = .Offset(Node1, 25)
    If NodeType = "M" Then GoTo NextNode

'Dump the A matrix of valid arcs into two arrays, one for ins and one for outs
    For Node2 = 1 To NumNodes
      InArray(Node2) = NodeMatrix(Node2, Node1)
      OutArray(Node2) = NodeMatrix(Node1, Node2)
    Next

'If we've made it this far, we've got a valid node to balance.  Next we create a balance
'equation for each part for Node1
  If FlowType = "P" Then

    For Part = 1 To NumParts

'Now start building the constraint, using six subs (one each for ins and outs this period,
'one each for ins and outs from/to previous/next period from the same node, and one each
'for ins and outs from/to previous/next period from a different node)

      For Period = 1 To NumPeriods
        If Node1 <> RemanNode Or (Node1 = RemanNode And_
        Worksheets("Data").Range("A2").Offset(Anchor1 + Part, 11) = "Y") Then
          Constraint = Constraint + 1
          Print #1, "Balance " & Node1 & " " & Constraint & ": ";
          If NodeType = "SDU" Or NodeType = "S" Then
            If Period = 1 Then
              Call BegInvArc
              Print #1, "+ " & Arc;
            End If
            Call SupplyArc
            Print #1, "+ " & Arc;
            Call SaveArc
```

```
            End If
          End If
'Now loop through all nodes in the applicable row and column of the A-matrix

          For Node2 = 1 To NumNodes


'First the arcs in...
'If there's a non-zero in any position, we need an arc from the i-node to Node1, unless the
'non-zero is in the diagonal of the matrix, which means we need an arc from Node1
period t-1 to
'Node1 period t.  If we have a 2, it means we need both

            If Node1 <> RemanNode Then

              If InArray(Node2) > 0 Then

                If Node2 <> Node1 And Node2 <> RemanNode Then
                  Call ArcIn
                  Print #1, "+ " & Arc;
                ElseIf Node2 <> Node1 And Node2 = RemanNode Then
                  If Worksheets("Data").Range("A2").Offset(Anchor1 + Part, 11) =_
                 "Y" Then
                    Call ArcIn
                    Print #1, "+ " & Arc;
                  End If
                ElseIf Node2 = Node1 Then
                  Call ArcFromLastPeriod
                  If Period > 1 Then
                    Print #1, "+ " & Arc;
                  End If
                End If

              End If

            ElseIf Node1 = RemanNode Then
              If InArray(Node2) > 0 Then
                If Worksheets("Data").Range("A2").Offset(Anchor1 + Part, 11) = "Y"_
                Then
                  Call ArcIn
                  Print #1, "+ " & Arc;
                End If
              End If
            End If

'Next the arcs out...
```

```
'Same logic applies as above...
          If Node1 <> RemanNode Then
            If OutArray(Node2) > 0 Then
              If Node2 = Node1 And Period < NumPeriods + 1 Then
                Call ArcToNextPeriod
                Call SaveArc
                Print #1, "- " & Arc;
              ElseIf Node2 <> Node1 And Node2 <> RemanNode Then
                Call ArcOut
                Call SaveArc
                Print #1, "- " & Arc;
              ElseIf Node2 <> 0 And Node2 = RemanNode Then
                If Worksheets("Data").Range("A2").Offset(Anchor1 + Part, 11) ="Y"
                Then
                  Call ArcOut
                  Call SaveArc
                  Print #1, "- " & Arc;
                End If
              End If
            End If
          ElseIf Node1 = RemanNode Then
            If OutArray(Node2) > 0 Then
              If Worksheets("Data").Range("A2").Offset(Anchor1 + Part, 11) = "Y"
              Then
                Call ArcOut
                Call SaveArc
                Print #1, "- " & Arc & " = 0"
              End If
            End If
          End If

        Next

        If NodeType = "DP" Or NodeType = "DU" Or NodeType = "DB" Or_
        NodeType = "SDU" Then

        'we've got a demand node

          If NodeType = "DP" Or NodeType = "DB" Then

            'we need a "DP" arc out, since we're filling part demand from this node
            If (PartsD = "Y" And Worksheets("Data").Range("A2")_
            .Offset(Anchor1 + Part, 11) = "Y") Or _
            (SellUsed = "Y" And Worksheets("Data").Range("A2")_
            .Offset(Anchor1 + Part, 12) = "Y") Then
```

```
            'but only if there's a market for it
              Call DemandArc
              Call SaveArc
              Print #1, "- ";
              Print #1, Arc;
            End If
            If NodeType = "DB" Then
              Call DemandArc
              Call SaveArc
              Print #1, "- ";
              Print #1, Arc2;
            End If

          ElseIf NodeType = "DU" Then
            Call DemandArc
            Call SaveArc
            Print #1, "- ";
            Print #1, Arc2;
          ElseIf NodeType = "SDU" Then
            Call DemandArc
            Call SaveArc
            Print #1, "- ";
            Print #1, Arc2;

          End If
        End If

        If ScrapOut = "Y" Then
          Call ScrapArc
          Call SaveArc
          Print #1, "- " & Arc;
        End If
        If Node1 <> RemanNode Then Print #1, "= 0"
      Next
    Next
  End If

  If FlowType = "C" Then
    For Core = 1 To NumCores
      For Period = 1 To NumPeriods
        Constraint = Constraint + 1
        Print #1, "Balance " & Node1 & " " & Constraint & ": ";
        For Node2 = 1 To NumNodes
          If InArray(Node2) > 0 Then
            Part = Core
            If NodeType = "S" Then
```

```
                If Period = 1 Then
                    Call BegInvArc
                    Print #1, "+ " & Arc;
                    Call SaveArc
                End If
                If CoreBuys = "Y" Then
                    Call CoreBuyArc
                    Print #1, "+ " & Arc;
                    Call SaveArc
                End If
            End If
            If Node2 <> Node1 Then
                Call ArcIn
                Print #1, "+ " & Arc;
            ElseIf Node2 = Node1 And Period > 1 Then
                Call ArcFromLastPeriod
                Print #1, "+ " & Arc;
            End If
        End If

'Next the arcs out...
'Same logic applies as above...

        If OutArray(Node2) > 0 Then
            Part = Core
            If Node2 = Node1 Then
                If Period < NumPeriods + 1 Then
                    Call ArcToNextPeriod
                    Call SaveArc
                    Print #1, "- " & Arc;
                End If
            ElseIf Node2 <> Node1 Then
                Call ArcOut
                Call SaveArc
                Print #1, "- " & Arc;
            End If
        End If

      Next
      Print #1, "= 0"
    Next
  Next
 End If
NextNode:
 Next
End With
```

```
End Sub
```

The following sub applies only to the node where cores are disassembled to their constituent parts. It ensures that for every core that is disassembled (i.e. enters the node), the corresponding number of used parts are available (i.e. leave the node)

```
Sub Multipliers()

For i = 1 To NumNodes
   If Worksheets("Balanced Nodes").Range("A3").Offset(i, 23) = "M" Then
      Node1 = i
      Node2 = i + 1
   End If
Next

With Worksheets("Data").Range("A2")
   For Period = 1 To NumPeriods
      For Part = 1 To NumParts
         Constraint = Constraint + 1
         Print #1, "Multipliers " & Constraint & ": ";
         For Core = 1 To NumCores
            If PartsArray(Part, 5) = "X" Then
               QPA = .Offset(Part, 6)
               Call MultiplierArcIn
               Print #1, Arc;
            End If
         Next
         Call MultiplierArcOut
         Call SaveArc
         Print #1, " - " & Arc & " = 0"
      Next
   Next
End With

End Sub
```

The following forces the correct amount of scrapped parts and usable parts

```
Sub Scrap()

With Worksheets("Data").Range("A2")
Node1 = 3
   For Period = 1 To NumPeriods
      For Part = 1 To NumParts
         Constraint = Constraint + 1
```

```
            Print #1, "Scrap " & Constraint & ": ";
            ScrapPct = .Offset(Anchor1 + Part, 19)
            Call ScrapArc
            Print #1, Arc & " - " & Arc2 & " >= 0"
        Next
    Next
End With

End Sub
```

The following sub is used only when individual parts are being remanufactured or repaired. It forces the correct percentage of repair.

```
Sub Reman()

With Worksheets("Data").Range("A2")
    For Period = 1 To NumPeriods
        For Part = 1 To NumParts
            If .Offset(Anchor1 + Part, 11) = "Y" Then
                Constraint = Constraint + 1
                Print #1, "Reman " & Constraint & ": ";
                RemanPct = .Offset(Anchor1 + Part, 13)
                Call RemanPctArc
                Print #1, Arc
            End If
        Next
    Next
End With

End Sub
```

The following sub establishes any disassembly capacity constraints

```
Sub Capacity()

    For Period = 1 To NumPeriods
        Constraint = Constraint + 1
        Print #1, "Capacity " & Constraint & ": ";
        For Core = 1 To NumCores
            Call CapacityArc
            Print #1, " + " & Arc;
        Next
    Print #1, " <= " & CapacityLimit
    Next

End Sub
```

The following sub builds balance constraints for demand nodes

```
Sub DemandNodes()

With Worksheets("Balanced Nodes").Range("A3")

'The following nested loops build balance constraints for each demand node by part and
'period, respectively.

If PartsD = "Y" Or SellUsed = "Y" Then
Demand = 0
   For Part = 1 To NumParts
      For Period = 1 To NumPeriods
         For Node1 = 1 To NumNodes
            'First, we check to see if Node1 satisfies parts demand
            NodeType = .Offset(Node1, 23)
               If NodeType = "DP" Or NodeType = "DB" Then
                  Call DemandArc
                  DemandConstraint = DemandConstraint & "+ " & Arc
               End If
         Next
         If (PartsD = "Y" And Worksheets("Data").Range("A2")_
         .Offset(Anchor1 + Part, 11) = "Y") Or _
         (SellUsed = "Y" And Worksheets("Data").Range("A2")_
         .Offset(Anchor1 + Part, 12) = "Y") Then
            Constraint = Constraint + 1
            Print #1, "Part Demand " & Constraint & " : " & DemandConstraint & " >= 0"
            Constraint = Constraint + 1
            Print #1, "Part Demand " & Constraint & " : " & DemandConstraint & " <= "_
            & (2 * CapacityLimit)
         Else
            Constraint = Constraint + 1
            Print #1, "Part Demand " & Constraint & " : " & DemandConstraint & " = 0"
         End If
         DemandConstraint = ""
      Next
   Next
End If

If UnitsD = "Y" Then
   For Part = 1 To NumParts
      For Period = 1 To NumPeriods
         Demand = DemandArray(NumCores + 1, Part, Period)
         For Node1 = 1 To NumNodes
```

```
            'First, we check to see if Node1 satisfies parts demand
            NodeType = .Offset(Node1, 23)
               If NodeType = "DU" Or NodeType = "SDU" Or NodeType = "DB" Then
                  Call DemandArc
                  DemandConstraint = DemandConstraint & "+ " & Arc2
               End If
         Next
         Constraint = Constraint + 1
         Print #1, "Unit Demand " & Constraint & " : " & DemandConstraint & " = "_
         & Demand
         DemandConstraint = ""
      Next
   Next
End If
End With

End Sub
```

The following sub builds balance constraints for supply nodes.

```
Sub SupplyNodes()

'This sub ensures the beginning inventory of cores is input into the system
With Worksheets("File Generator").Range("A10")
   Node1 = 1
   NodeType = "S"
   FlowType = "C"
   For Core = 1 To NumCores
      For Period = 1 To NumPeriods
         BegInv = ReceiptArray(Core, Period)
         Constraint = Constraint + 1
         Print #1, "Supply " & Constraint & ": ";
         Call BegInvArc
         Print #1, Arc & " = " & BegInv
      Next
   Next
End With

End Sub
```

The next sub builds integer constraints (optional – the LP approximation is much faster, and was shown in pilot tests to yield comparable results)

```
Sub Integers()

With Worksheets("Balanced Nodes").Range("W17")
```

```
For i = 1 To 10
   If .Offset(i, 0) <> "" Then
      Node1 = .Offset(i, 0)
      Destination = .Offset(i, 2)
      For Period = 1 To NumPeriods
         For Part = 1 To NumParts

            Call IntegerArc
            Print #1, Arc

         Next
      Next
   End If
Next
End With

End Sub
```

The following sub creates non-negativity constraints

```
Sub NonNegativity()

With Worksheets("Balanced Nodes").Range("A3")

'The following nested loops build balance constraints for each node by part and period,
'respectively.

   For Node1 = 1 To NumNodes

'Dump the A matrix of valid arcs into two arrays, one for ins and one for outs
      For Node2 = 1 To NumNodes
         InArray(Node2) = NodeMatrix(Node2, Node1)
         OutArray(Node2) = NodeMatrix(Node1, Node2)
      Next

'Now we enforce non-negativity for all ins and outs to Node1 across all periods and parts

      For Part = 1 To NumParts


'Now start building the constraint, using six subs (one each for ins and outs this period,
'one each for ins and outs from/to previous/next period from the same node, and one each
'for ins and outs from/to previous/next period from a different node)

         For Period = 1 To NumPeriods
```

```
'Now loop through all nodes in the applicable row and column of the A-matrix

            For Node2 = 1 To NumNodes

'Only need to do the arcs out, so we don't duplicate

               If OutArray(Node2) > 0 Then

                  If Node2 = Node1 And Period < NumPeriods Then
                     Constraint = Constraint + 1
                     Print #1, "NonNegativity " & Constraint & ": ";
                     Call ArcToNextPeriod
                     Print #1, Arc & ">= 0"
                  ElseIf i <> Node1 Then
                     Constraint = Constraint + 1
                     Print #1, "NonNegativity " & Constraint & ": ";
                     Call ArcOut
                     Print #1, Arc & ">= 0"
                  End If
               End If
               If OutArray(Node2) = 2 And i > Node1 And Period < NumPeriods Then
                  Constraint = Constraint + 1
                  Print #1, "NonNegativity " & Constraint & ": ";
                  Call ArcNextAnotherNode
                  Print #1, Arc & ">= 0"
               End If
            Next
         Next
      Next
   Next
End With

End Sub
```

The next set of subs simply fills the various arrays used to create the problem file

```
Sub FillReceiptArray()

With Worksheets("File Generator").Range("D14")
   For Core = 1 To NumCores
      For Period = 1 To NumPeriods
         ReceiptArray(Core, Period) = .Offset(Core, Period)
      Next
   Next
End With
```

```
End Sub

Sub FillPartsArray()

With Worksheets("Data").Range("A2")
   'Write the parts list to the parts array
   For Core = 1 To NumCores
      For Part = 1 To NumParts
         PartsArray(Part, 1) = .Offset(Anchor1 + Part, 7)
         PartsArray(Part, 2) = .Offset(Anchor1 + Part, 8)
         PartsArray(Part, 3) = .Offset(Anchor1 + Part, 19)
         PartsArray(Part, 4) = .Offset(Anchor1 + Part, 20)
         PartsArray(Part, 5) = .Offset(Anchor1 + Part, 21)
         PartsArray(Part, 6) = .Offset(Anchor1 + Part, 6)
      Next
   Next
End With

End Sub


Sub FillDemandArray()

'First we build a demand matrix to calculate and store the demand for all parts
'corresponding to the demands for all cores

For Core = 1 To NumCores
   For Part = 1 To NumParts
      For Period = 1 To NumPeriods
         If PartsArray(Part, 5) = "X" Then
            CoreDemand = CoreDemandArray(Core, Period)
            PartDemand = CoreDemand * PartsArray(Part, 6)
            DemandArray(Core, Part, Period) = PartDemand
         Else
            DemandArray(Core, Part, Period) = 0
         End If
      Next
   Next
Next

'Now we total the parts demand across all cores for each part in each period

   For Part = 1 To NumParts
      For Period = 1 To NumPeriods
         For Core = 1 To NumCores
            DemandArray(NumCores + 1, Part, Period) =_
```

```
            DemandArray(NumCores + 1, Part, Period) +_
            DemandArray(Core, Part, Period)
         Next
      Next
   Next

End Sub


Sub FillSupplyArray()

With Worksheets("File Generator").Range("A10")
   For Core = 1 To NumCores
      SupplyArray(Core) = .Offset(Core, 1)
   Next
End With

End Sub


Sub FillNodeMatrix()

With Worksheets("Balanced Nodes").Range("A3")
   For Node1 = 1 To NumNodes
      For Node2 = 1 To NumNodes
         NodeMatrix(Node1, Node2) = .Offset(Node1, Node2)
      Next
   Next
End With

End Sub


Sub FillCoreDemand()

With Worksheets("File Generator").Range("D10")
   For Core = 1 To NumCores
      For Period = 1 To NumPeriods
         CoreDemandArray(Core, Period) = .Offset(Core, Period)
      Next
   Next
End With

End Sub
```

```
Sub FillRemanPrices()

With Worksheets("File Generator").Range("C10")
   For Core = 1 To NumCores
      RemanPrices(Core) = .Offset(Core, 0)
   Next
End With

End Sub
```

The final set of subs build the various types of arcs required by the main program

```
Sub ArcIn()
'Builds an arc into Node1 from another node, same period
   If FlowType = "P" Then
      Arc = "X" & Node2 & "_" & Period & "_" & Node1 & "_" & Period & "_"_
      & Part & " "
   ElseIf FlowType = "C" Then
      Arc = "X" & Node2 & "_" & Period & "_" & Node1 & "_" & Period & "_C"_
      & Part & " "
   End If
End Sub


Sub BegInvArc()
'This sub establishes the beginning inventory
   If FlowType = "P" Then
      Arc = "XPI_" & Period & "_" & Node1 & "_" & Period & "_" & Part & " "
   ElseIf FlowType = "C" Then
      Arc = "XCS_" & Period & "_" & Node1 & "_" & Period & "_C" & Core & " "
   End If
End Sub

Sub ArcOut()
'Builds an arc out of Node1 to another node, same period
   If FlowType = "P" Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node2 & "_" & Period & "_"_
      & Part & " "
   ElseIf FlowType = "C" Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node2 & "_" & Period & "_C"_
      & Part & " "
   End If
End Sub
```

```
Sub DemandArc()
'Builds a demand arc out of a demand node
  If NodeType = "DP" Then
    Arc = "X" & Node1 & "_" & Period & "_DP_" & Period & "_" & Part & " "
  ElseIf NodeType = "DU" Or NodeType = "SDU" Then
    Arc2 = "X" & Node1 & "_" & Period & "_DU_" & Period & "_" & Part & " "
  ElseIf NodeType = "DB" Then
    Arc = "X" & Node1 & "_" & Period & "_DP_" & Period & "_" & Part & " "
    Arc2 = "X" & Node1 & "_" & Period & "_DU_" & Period & "_" & Part & " "
  End If
End Sub

Sub SupplyArc()
'Builds a supply arc
  If NodeType = "S" Then
    Arc = "XCS_" & Period & "_" & Node1 & "_" & Period & "_C" & Core & " "
  ElseIf NodeType = "SDU" Then
    Arc = "XPS_" & Period & "_" & Node1 & "_" & Period & "_" & Part & " "
  End If
End Sub

Sub CoreBuyArc()
'Builds an arc for cores bought on the market
  Arc = "XBC_" & Period & "_" & Node1 & "_" & Period & "_C" & Core & " "
End Sub

Sub ArcToNextPeriod()
'Builds an arc from Node1, this period to Node1, next period (inventory holdover)
  If Period < NumPeriods Then
    If FlowType = "P" Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_" & Period + 1 & "_"_
      & Part & " "
    ElseIf FlowType = "C" Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_" & Period + 1 & "_C"_
      & Part & " "
    End If
  ElseIf Period = NumPeriods Then
    If FlowType = "P" Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_1_" & Part & " "
    ElseIf FlowType = "C" Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_1_C" & Part & " "
    End If
  End If
End Sub
```

```
Sub ArcFromLastPeriod()
'Builds an arc from Node1 last period to Node1, this period
   If Period > 1 Then
      If FlowType = "P" Then
         Arc = "X" & Node1 & "_" & Period - 1 & "_" & Node1 & "_" & Period & "_"_
         & Part & " "
      ElseIf FlowType = "C" Then
         Arc = "X" & Node1 & "_" & Period - 1 & "_" & Node1 & "_" & Period & "_C"_
         & Part & " "
      End If
   ElseIf Period = 1 Then
      If FlowType = "P" Then
         Arc = "X" & Node1 & "_" & NumPeriods & "_" & Node1 & "_" & Period_
         & "_" & Part & " "
      ElseIf FlowType = "C" Then
         Arc = "X" & Node1 & "_" & NumPeriods & "_" & Node1 & "_" & Period_
         & "_C" & Part & " "
      End If
   End If
End Sub


Sub ArcLastAnotherNode()
'Builds an arc into Node1 from another node, last period
   Arc = "X" & i & "_" & Period - 1 & "_" & Node1 & "_" & Period & "_" & Part & " "
End Sub


Sub ArcNextAnotherNode()
'Builds an arc out of Node1 to another node, next period
   Arc = "X" & Node1 & "_" & Period & "_" & i & "_" & Period + 1 & "_" & Part & " "
End Sub


Sub MultiplierArcIn()
'Builds a constraint to ensure the core is disassembled into appropriate parts
   Arc = " + " & QPA & "X" & Node1 - 1 & "_" & Period & "_" & Node1_
   & "_" & Period & "_C" & Core
End Sub

Sub MultiplierArcOut()
'Builds the multiplier arcs out of a multiplier node
   Arc = "X" & Node1 & "_" & Period & "_" & Node2 & "_" & Period & "_" & Part
End Sub
```

```
Sub ScrapArc()
'Builds a constraint to ensure the correct percentage of parts are sent to scrap
   Arc = "X" & Node1 & "_" & Period & "_Scrap_" & Period & "_" & Part
   Arc2 = ScrapPct & "X" & Node1 - 1 & "_" & Period & "_" & Node1_
   & "_" & Period & "_" & Part
End Sub


Sub RemanPctArc()
'Builds a constraint to ensure the correct percentage of parts are remanufactured
   Arc = "X3_" & Period & "_4_" & Period & "_" & Part & " - " & _
   RemanPct & "X2_" & Period & "_3_" & Period & "_" & Part & " >= 0"
End Sub


Sub CapacityArc()
'Builds an arc to build the capacity constraints
   Arc = "X1_" & Period & "_2_" & Period & "_C" & Core
End Sub


Sub DisassemblyArc()
'Builds disassembly arcs for the objective function
   Arc = "X1_" & Period & "_2_" & Period & "_C" & Core
End Sub


Sub RemanArc()
'Builds reman arcs for the objective function
   If Period <> NumPeriods Then
      Arc = "X3_" & Period & "_4_" & Period & "_" & Part
   Else
      Arc = "X3_" & Period & "_4_1_" & Part
   End If
End Sub


Sub NewPartArc()
'Builds new part arcs for the objective function
   Arc = "XPS_" & Period & "_5_" & Period & "_" & Part
End Sub
```

```
Sub HoldingCostArc()
'Builds holding cost arcs for parts carried over to next period
If Node1 = 1 Then
   If Period <> NumPeriods Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_" & Period + 1_
      & "_C" & Core
   Else
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_1_C" & Core
   End If
Else
   If Period <> NumPeriods Then
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_" & Period + 1 & "_" &
Part
   Else
      Arc = "X" & Node1 & "_" & Period & "_" & Node1 & "_1_" & Part
   End If
End If
End Sub


Sub PartProfitArc()
'Builds profit arcs for sale of individual used parts
   Arc = "X5_" & Period & "_DP_" & Period & "_" & Part
End Sub


Sub UnitProfitArc()
'Builds profit arcs for sales of remanufactured units
   Arc = Profit & "X5_" & Period & "_DU_" & Period & "_1 + " & Profit & "X6_" & _
   Period & "_DU_" & Period & "_1"
End Sub


Sub IntegerArc()
'Builds an arc to include in the integer declaration section
   Arc = "X" & Node1 & "_" & Period & "_" & Destination & "_" & Period & "_" & Part
End Sub

Sub SaveArc()
'Saves each arc created in an array called ArcArray, for use in bounding and integer
constraints
ArcCounter = ArcCounter + 1
ReDim Preserve ArcArray(ArcCounter)
ArcArray(ArcCounter) = Arc
```

```
End Sub
```

2. Deterministic Solution Import and Format

The previous program created a text file for the problem(s), which were imported by CPLEX for solution. The solution files were generated in text format. The following program imports the solution files and formats them to a convenient form.

Variable declaration

```
Public Qd As Integer, P1 As Integer, P2 As Integer, P3 As Integer, P4 As Integer
Public P5 As Integer, P6 As Integer, P7 As Integer, P8 As Integer, P9 As Integer
Public P10 As Integer
Public OVal As Double
Public Problem As Integer
```

The "Main" sub calls, for each problem, four subs that import the data, delete unwanted rows, get the solution values of interest, and transfer them to a separate sheet, respectively.

```
Sub Main()
      For Problem = 1 To 16
         Worksheets("Import").Cells.ClearContents
         Call ImportData
         Call DeleteRows
         Call GetValues
         Call Transfer
      Next
End Sub

Sub ImportData()
        Dim r As Integer, ImpRng As Variant, data As String
        Worksheets("Import").Range("A1").Select
        Set ImpRng = ActiveCell
        Open "c:\d\s\" & Problem & ".txt" For Input As #1
        r = 0
        Do Until EOF(1)
           Line Input #1, data
           ActiveCell.Offset(r, 0) = data
           r = r + 1
        Loop
        Close #1
End Sub
```

```
Sub DeleteRows()
        Dim Datarow As String, i As Integer, Startline As Integer, Done As Boolean
        Dim j As Integer, ToFind As String, Objective As String, ObjVal As String
        ToFind = "SECTION 2"
        Done = False
        i = -1

        With Worksheets("Import").Range("A1")
          Do
            i = i + 1
            Datarow = .Offset(i, 0)
            Done = ISLIKE(Datarow, " SECTION 2*")
          Loop Until Done
          Startline = i
          i = 0

          Do
            i = i + 1
            Datarow = .Offset(i, 0)
            Done = ISLIKE(Datarow, " OBJECTIVE*")
          Loop Until Done
          Objective = .Offset(i, 0)
          ObjVal = Mid(Objective, 19, 8)
          For j = 1 To Startline + 4
            ActiveSheet.Range("A1").Cells.Activate
            ActiveCell.Delete
          Next

        'Format the text into columns
          Columns("A:A").Select
          Selection.TextToColumns Destination:=Range("A1"),
        DataType:=xlFixedWidth, _
            FieldInfo:=Array(Array(0, 1), Array(8, 1), Array(25, 1), Array(32, 1),
        Array(48, 1), _
            Array(68, 1), Array(82, 1), Array(96, 1)), TrailingMinusNumbers:=True
        'Insert two rows
          Rows("1:1").Select
          Selection.Insert Shift:=xlDown
          Selection.Insert Shift:=xlDown
        With Worksheets("Import").Range("A1")
          .Offset(0, 0) = "OBJ"
          .Offset(0, 1) = ObjVal
          .Offset(1, 0) = "NUMBER"
          .Offset(1, 1) = "ARC"
```

```
            .Offset(1, 2) = "AT"
            .Offset(1, 3) = "ACTIVITY"
            .Offset(1, 4) = "COST"
            .Offset(1, 5) = "LL"
            .Offset(1, 6) = "UL"
            .Offset(1, 7) = "RC"
        End With
        End With
End Sub


Function ISLIKE(text As String, Pattern As String) As Boolean
  If text Like Pattern Then ISLIKE = True _
    Else ISLIKE = False
End Function


Sub GetValues()

Dim i As Integer
Dim NumLines As Integer
Qd = 0
P1 = 0
P2 = 0
P3 = 0
P4 = 0
P5 = 0
P6 = 0
P7 = 0
P8 = 0
P9 = 0
P10 = 0
With Worksheets("Import").Range("A1")
  OVal = .Offset(0, 1)
  NumLines = Range(.Offset(2, 0), .End(xlDown)).Rows.Count
  For i = 1 To NumLines
    If .Offset(i + 1, 1) = "XBC_1_1_1_C1" Then Qd = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_1" Then P1 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_2" Then P2 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_3" Then P3 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_4" Then P4 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_5" Then P5 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_6" Then P6 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_7" Then P7 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_8" Then P8 = .Offset(i + 1, 3)
    If .Offset(i + 1, 1) = "XPS_1_5_1_9" Then P9 = .Offset(i + 1, 3)
```

```
        If .Offset(i + 1, 1) = "XPS_1_5_1_10" Then P10 = .Offset(i + 1, 3)
    Next
End With

End Sub


Sub Transfer()

Dim i As Integer

With Worksheets("Results").Range("A1")
.Offset(Problem, 0) = Problem
.Offset(Problem, 11) = OVal
.Offset(Problem, 12) = Qd
.Offset(Problem, 1) = P1
.Offset(Problem, 2) = P2
.Offset(Problem, 3) = P3
.Offset(Problem, 4) = P4
.Offset(Problem, 5) = P5
.Offset(Problem, 6) = P6
.Offset(Problem, 7) = P7
.Offset(Problem, 8) = P8
.Offset(Problem, 9) = P9
.Offset(Problem, 10) = P10
End With

End Sub
```

3. Safety Stock

The final program takes the deterministic LP solution and adds safety stock to achieve the target service level. The result is the final solution, which is the one compared with the SIPR solutions in the main experiment.

```
Sub SafetyStock()
```

Variable declarations

```
Dim QtyArray() As Integer
Dim NumProbs As Integer
Dim i As Integer, j As Integer, k As Integer
```

```
Dim Problem As Integer
Dim NumParts As Integer
Dim Hurdle As Double, TSL As Double, SL As Double
Dim Done As Boolean
Dim Cost As Double
Dim PartArray() As Single
Dim Anchor1 As Integer, Anchor2 As Integer
Dim StartTime As Double, EndTime As Double
Dim SubSet As Integer, Subset2 As Integer
Dim SortCost() As Double
Dim First As Integer, Last As Integer, Lowest() As Double
Dim Level As Integer
Dim SD As Double

'First grab the deterministic solutions from the Results worksheet
'and load them into the Quantity Array

With Worksheets("Results").Range("A1")
    NumProbs = Range(.Offset(1, 0), .End(xlDown)).Rows.Count
    ReDim QtyArray(NumProbs, 11)
    ReDim Lowest(2)

    For i = 1 To NumProbs
        For j = 1 To 10
            QtyArray(i, j) = .Offset(i, j)
        Next
        QtyArray(i, 11) = .Offset(i, 12)
    Next
End With

'Initialize the anchors that mark the problem
    Anchor1 = 0
    Anchor2 = 0

'Now load one problem at a time and calculate safety levels
For Level = 1 To NumLevels
    Anchor1 = 0
    Anchor2 = 0
    SD = Worksheets("Calc Sheet").Range("A1").Offset(Level + 26, 0)
    Worksheets("Calc Sheet").Range("A1").Offset(15, 1) = SD

For Problem = 1 To NumProbs
StartTime = Timer
With Worksheets("Problems").Range("A2")

'First count the number of parts in the problem and reset the anchor
```

```
'for the next problem

   Anchor1 = Anchor2
      j = 0
      Do
         j = j + 1
         If .Offset(Anchor1 + j, 1) <> "" Then
            NumParts = .Offset(Anchor1 + j, 1)
         End If
      Loop Until .Offset(Anchor1 + j, 1) = ""
      Anchor2 = Anchor1 + j

'Now redimension the parts and solution arrays for the number of parts
      ReDim PartArray(NumParts, 2)
      ReDim SortCost(NumParts, 2)
'Read in the parts data
      For i = 1 To NumParts
         PartArray(i, 1) = .Offset(Anchor1 + i, 2)
         PartArray(i, 2) = .Offset(Anchor1 + i, 3)
      Next
End With

With Worksheets("Calc Sheet").Range("A1")

'Write problem yields and costs to calculation worksheet
   .Offset(24, 0) = Problem
   For i = 1 To NumParts
      .Offset(i + 2, 1) = PartArray(i, 1)
      .Offset(i + 2, 3) = PartArray(i, 2)
   Next
'Fill in zeros if less than 10 parts
   If NumParts < 10 Then
      For i = NumParts + 1 To 10
         .Offset(i + 2, 1) = 0
         .Offset(i + 2, 3) = 0
         .Offset(i + 2, 5) = 10000
      Next
   End If
'Sort the parts in order of ascending costs and store in SortCost array
   For i = 1 To NumParts
      SortCost(i, 1) = .Offset(i + 2, 3)
      SortCost(i, 2) = i
   Next
   First = 1
   Last = NumParts
   For i = First To Last - 1
```

```
        For j = i + 1 To Last
          If SortCost(i, 1) > SortCost(j, 1) Then
            Lowest(1) = SortCost(j, 1)
            Lowest(2) = SortCost(j, 2)
            SortCost(j, 1) = SortCost(i, 1)
            SortCost(j, 2) = SortCost(i, 2)
            SortCost(i, 1) = Lowest(1)
            SortCost(i, 2) = Lowest(2)
          End If
        Next
      Next

'Now write the solution to the Calculation sheet
        If Problem <= 8 Then
          For j = 1 To 10
            .Offset(j + 2, 5) = QtyArray(Problem, j)
          Next
        Else
          For j = 1 To 5
            .Offset(j + 2, 5) = QtyArray(Problem, j)
            .Offset(j + 7, 5) = 10000
          Next
        End If
        .Offset(2, 5) = QtyArray(Problem, 11)

'Now calculate the necessary part service levels to achieve the overall TSL
        TSL = .Offset(17, 2)
        Hurdle = TSL ^ (1 / NumParts)
        SubSet = NumParts

        Do
          Done = True
          Subset2 = SubSet
          SubSet = NumParts
              For j = 1 To NumParts
                If .Offset(j + 2, 8) >= Hurdle Then
                  SubSet = SubSet - 1
                  Done = False
                End If
              Next
              If Subset2 = SubSet Then Done = True
              If SubSet = 0 Then GoTo Cya

              Hurdle = TSL ^ (1 / SubSet)
        Loop Until Done
```

```
'Increment the quantity for each part until it reaches its hurdle
    For i = 1 To NumParts
      j = SortCost(i, 2)
        Done = False
        Do
          If .Offset(15, 7) >= .Offset(17, 1) Or .Offset(j + 2, 8) >= Hurdle Then
            Done = True
          ElseIf .Offset(j + 2, 8) < Hurdle Then
            .Offset(j + 2, 5) = .Offset(j + 2, 5) + 1
          End If
        Loop Until Done
    Next
    EndTime = Timer

Cya:

'Write new solution to QtyArray
    For j = 1 To 10
      QtyArray(Problem, j) = .Offset(j + 2, 5)
    Next
    Cost = .Offset(14, 6)
    SL = .Offset(18, 1)

End With

'And then to the SS Results worksheet
  With Worksheets("SS Results").Range("A1")
    For j = 1 To NumParts
      .Offset((3 * Level) + Problem - 3, j + 1) = QtyArray(Problem, j)
    Next
    .Offset((3 * Level) + Problem - 3, 0) = Problem
    .Offset((3 * Level) + Problem - 3, 1) = Level
    .Offset((3 * Level) + Problem - 3, 7) = Cost
    .Offset((3 * Level) + Problem - 3, 8) = QtyArray(Problem, 11)
    .Offset((3 * Level) + Problem - 3, 9) = SL
    .Offset((3 * Level) + Problem - 3, 10) = EndTime - StartTime
  End With

Next
Next
Beep
End Sub
```

APPENDIX C – SIPR Code

Unlike ReNet, SIPR uses a single program to attain a solution.

Variable declaration

```
Dim a As Integer, b As Integer, c As Integer, d As Integer, e As Integer
Dim r As Integer, s As Integer, t As Integer, u As Integer, v As Integer
Dim Counter As Integer
Dim i As Integer, j As Integer, n As Integer, k As Integer, m As Integer
Dim ServiceLevel() As Double, ServiceTarget As Double
Dim PartArray(), SolutionArray() As Double
Dim Part As Integer
Dim Yield As Double, Cost As Double
Dim Quantity As Integer, PropQty As Integer
Dim Shortage As Double, ExpShort As Double
Dim TotShortage1 As Double, TotShortage2 As Double
Dim Solution As Integer, Solution2 As Integer
Dim Comparison() As Double
Dim NumProbs As Integer, Problem As Integer, NumParts As Integer
Dim Anchor1 As Integer, Anchor2 As Integer
Dim Range1 As Integer, Range2 As Integer
Dim Best As Integer
Dim StartTime As Double, EndTime As Double
Dim SL As Double
Dim Qd() As Double
Dim AllDone As Boolean
Dim MovingAvg1 As Double, MovingAvg2 As Double
Dim Trend As Integer
Dim FinalStep() As Integer
Dim Total As Integer
Dim Least As Double
Dim Check As Boolean
Dim BackUp() As Integer
Dim Once As Integer
Dim BackUpCost As Double
Dim Replace As Boolean
```

Sub "SIPR" calls the other subs for each problem in order

```
Sub SIPR()
        Application.ScreenUpdating = False
        Call CountProbs
        For Problem = 1 To NumProbs
           Call Initialize
           Call CountParts
           Call ArrayMgmt
           Call WriteValues
           Call SetRange
           Call QdLoop
        Next
End Sub
```

The following sub simply counts the number of problems to be solved

```
Sub CountProbs()
   With Worksheets("Problems").Range("A2")
   'First count the number of problems in the problem set
      For i = 0 To 500
         If .Offset(i, 0) > 0 Then
            NumProbs = .Offset(i, 0)
         End If
      Next
   'Initialize the anchors that mark the problem
      Anchor1 = 0
      Anchor2 = 0
   End With
End Sub
```

Sub "Initialize" initializes those variables that must be initialized for each problem

```
Sub Initialize()
   StartTime = Timer
   Trend = 0
   Anchor1 = Anchor2
   MovingAvg1 = 100000
End Sub
```

The following sub counts the number of parts in the current problem

```
Sub CountParts()
   With Worksheets("Problems").Range("A2")
      'First count the number of parts in the problem and reset the anchor
      'for the next problem
      j = 0
      Do
         j = j + 1
         If .Offset(Anchor1 + j, 1) <> "" Then
            NumParts = .Offset(Anchor1 + j, 1)
         End If
      Loop Until .Offset(Anchor1 + j, 1) = ""
      Anchor2 = Anchor1 + j
   End With
End Sub
```

The following sub redimensions the arrays to be used in the solution technique

```
Sub ArrayMgmt()
   With Worksheets("Problems").Range("A2")
      'Now redimension the parts and solution arrays for the number of parts
      ReDim PartArray(NumParts, 2)
      ReDim SolutionArray(NumParts)
      ReDim ServiceLevel(NumParts)
      ReDim Comparison(NumParts + 3, 1)
      ReDim Qd(3)
      For i = 1 To 3
         Qd(i) = 30000
      Next
      'Read in the parts data
      For i = 1 To NumParts
         PartArray(i, 1) = .Offset(Anchor1 + i, 2)
         PartArray(i, 2) = .Offset(Anchor1 + i, 3)
      Next
      Range1 = .Offset(Anchor1, 4)
   End With
End Sub
```

The following sub writes the problem parameters to the calculation sheet prior to solving

```
Sub WriteValues()
   'Write the values to the calculation sheet
   With Worksheets("Calc Sheet").Range("A1")
```

```
      .Offset(24, 0) = Problem
      For i = 1 To NumParts
         .Offset(i + 2, 1) = PartArray(i, 1)
         .Offset(i + 2, 3) = PartArray(i, 2)
      Next
   'Fill in zeros if less than 10 parts
      If NumParts < 10 Then
         For i = NumParts + 1 To 10
            .Offset(i + 2, 1) = 0
            .Offset(i + 2, 3) = 0
            .Offset(i + 2, 5) = 10000
         Next
      End If
   End With
End Sub
```

The following sub sets the starting value for Q^d (user input)

```
Sub SetRange()
'Set the search range for Qd
   With Worksheets("Calc Sheet").Range("A1")
      Range1 = .Offset(16, 1)
   End With
End Sub
```

The "QdLoop" sub solves the optimal set of $\{Q_p\}$ for the range of Q^d

```
Sub QdLoop()
With Worksheets("Calc Sheet").Range("A1")
'Now loop through range of Qd values, solving Qp for each
   AllDone = False
   n = Range1 - 1
   Do Until AllDone
      n = n + 1
      Once = 0
      ReDim Preserve Comparison(NumParts + 3, n - Range1 + 1)
      ReDim FinalStep(NumParts, 2)
      ReDim BackUp(NumParts)
      .Offset(2, 5) = n
      'Reset part order quantities to zero
      For i = 1 To NumParts
         .Offset(i + 2, 5) = 0
      Next

'     Call StartMin
```

```
        Call StartAvg
'       Call StartBack
        Call StartingSL
        Call Optimize
'       Call CheckOnes
        Call RevisedCheck
        Call SaveSolution
        Call TrendCheck
    Loop
    Call ChooseBest
    Call WriteSolution
End With
End Sub

Sub StartMin()
    With Worksheets("Calc Sheet").Range("A1")
'Increment each part quantity until the service level is 1
        For i = 1 To NumParts
            ServiceLevel(i) = .Offset(i + 2, 8)
        Next
        For i = 1 To NumParts
            Do Until ServiceLevel(i) = 0
                .Offset(i + 2, 5) = .Offset(i + 2, 5) + 1
                ServiceLevel(i) = .Offset(i + 2, 8)
            Loop
        Next
    End With
End Sub

Sub StartAvg()
    With Worksheets("Calc Sheet").Range("A1")
'Set initial order qtys to the expected value needed
        For i = 1 To NumParts
            If .Offset(i + 2, 5) < Round((.Offset(16, 1) - (.Offset(i + 2, 1) * _
            .Offset(2, 5))) - 0.5, 0) Then
                .Offset(i + 2, 5) = Round((.Offset(16, 1) - (.Offset(i + 2, 1) * _
                .Offset(2, 5))) - 0.5, 0)
            End If
        Next
    End With
End Sub

Sub StartingSL()
    With Worksheets("Calc Sheet").Range("A1")
    'Take initial read of service level and cost with part quantities seeded
        ServiceTarget = .Offset(17, 1)
```

```
        Cost = .Offset(14, 6)
        For i = 1 To NumParts
            ServiceLevel(i) = .Offset(i + 2, 8)
        Next
    End With
End Sub

Sub StartBack()
    With Worksheets("Calc Sheet").Range("A1")
        Do
            For i = 1 To NumParts
                If .Offset(i + 2, 5) > 0 Then
                    .Offset(i + 2, 5) = .Offset(i + 2, 5) - 1
                    SolutionArray(i) = (Cost - .Offset(14, 6)) / (ServiceLevel(i) - _
                    .Offset(i + 2, 8))
                    .Offset(i + 2, 5) = .Offset(i + 2, 5) + 1
                Else
                    SolutionArray(i) = 0
                End If
            Next
            Solution = 1
            For i = 2 To NumParts
                If SolutionArray(i) > SolutionArray(Solution) Then
                    Solution = i
                End If
            Next
            .Offset(Solution + 2, 5) = .Offset(Solution + 2, 5) - 1
            For i = 1 To NumParts
                ServiceLevel(i) = .Offset(i + 2, 8)
            Next
            SL = .Offset(15, 8)
        Loop Until (SL <= .Offset(19, 1))
    End With
End Sub

Sub Optimize()
Counter = 1
With Worksheets("Calc Sheet").Range("A1")
'Now increase each part by one unit and store the bang per buck in solution array
    Do
        For i = 1 To NumParts
            FinalStep(i, 2) = FinalStep(i, 1)
            FinalStep(i, 1) = .Offset(i + 2, 5)
            .Offset(i + 2, 5) = .Offset(i + 2, 5) + 1
            SolutionArray(i) = (.Offset(i + 2, 8) - ServiceLevel(i)) / (.Offset(14, 6) - Cost)
            .Offset(i + 2, 5) = .Offset(i + 2, 5) - 1
```

```
            Next
            Solution = 1
            k = 0
            For i = 2 To NumParts
               If SolutionArray(i) > SolutionArray(Solution) Then
                  Solution = i
               ElseIf SolutionArray(i) = SolutionArray(Solution) Then
                  k = k + 1
               End If
            Next
            Total = 0
            Least = 10000

            For i = 1 To NumParts
               ServiceLevel(i) = .Offset(i + 2, 8)
            Next
            SL = .Offset(15, 8)
            Cost = .Offset(14, 6)
            Least = Cost
            Worksheets("Test").Range("A1").Offset(Counter, 0) = .Offset(3, 5)
            Worksheets("Test").Range("A1").Offset(Counter, 1) = .Offset(4, 5)
            Worksheets("Test").Range("A1").Offset(Counter, 2) = .Offset(5, 5)
            Worksheets("Test").Range("A1").Offset(Counter, 3) = .Offset(6, 5)
            Worksheets("Test").Range("A1").Offset(Counter, 4) = .Offset(7, 5)
            Worksheets("Test").Range("A1").Offset(Counter, 5) = Cost
            Worksheets("Test").Range("A1").Offset(Counter, 6) = SL
            Worksheets("Test").Range("A1").Offset(Counter, 7) = .Offset(18, 1)
            Counter = Counter + 1
         Loop Until (SL >= ServiceTarget)
      For i = 1 To NumParts
         BackUp(i) = .Offset(i + 2, 5)
      Next
      End With
      End Sub


      Sub CheckOnes()

      With Worksheets("Calc Sheet").Range("A1")
      'Before saving the solution, check back to see if any of the BackUp Solutions are better
         For j = 1 To 2
            For i = 1 To NumParts
               .Offset(i + 2, 5) = FinalStep(i, j)
            Next
      'Check for single-steps first
            For i = 1 To NumParts
```

```
            .Offset(i + 2, 5) = .Offset(i + 2, 5) + 1
            If .Offset(15, 8) >= .Offset(17, 1) Then
               If .Offset(14, 6) < Least Then
                  For k = 1 To NumParts
                     BackUp(k) = .Offset(k + 2, 5)
                  Next
                  BackUpCost = .Offset(14, 6)
                  Least = BackUpCost
               End If
            End If
            .Offset(i + 2, 5) = .Offset(i + 2, 5) - 1
         Next
'Now check for two-steps
         For i = 1 To NumParts
            For k = 1 To NumParts
               .Offset(i + 2, 5) = .Offset(i + 2, 5) + 1
               .Offset(k + 2, 5) = .Offset(k + 2, 5) + 1
               If .Offset(15, 8) >= .Offset(17, 1) Then
                  If .Offset(14, 6) < Least Then
                     For m = 1 To NumParts
                        BackUp(m) = .Offset(m + 2, 5)
                     Next
                     BackUpCost = .Offset(14, 6)
                     Least = BackUpCost
                  End If
               End If
            .Offset(i + 2, 5) = .Offset(i + 2, 5) - 1
            .Offset(k + 2, 5) = .Offset(k + 2, 5) - 1
            Next
         Next
      Next
If BackUpCost < Cost Then
   For i = 1 To NumParts
      .Offset(i + 2, 5) = BackUp(i)
      Cost = BackUpCost
   Next
End If
End With
End Sub


Sub RevisedCheck()
Replace = False
With Worksheets("Calc Sheet").Range("A1")
For i = 1 To NumParts
   .Offset(i + 2, 5) = BackUp(i)
```

```
   If .Offset(i + 2, 5) > 0 Then .Offset(i + 2, 5) = .Offset(i + 2, 5) - 1
Next
If NumParts > 5 Then
   For a = 0 To 1
      .Offset(3, 5) = .Offset(3, 5) + a
      For b = 0 To 1
         .Offset(4, 5) = .Offset(4, 5) + b
         For c = 0 To 1
            .Offset(5, 5) = .Offset(5, 5) + c
            For d = 0 To 1
               .Offset(6, 5) = .Offset(6, 5) + d
               For e = 0 To 1
                  .Offset(7, 5) = .Offset(7, 5) + e
                  For r = 0 To 1
                     .Offset(8, 5) = .Offset(8, 5) + r
                     For s = 0 To 1
                        .Offset(9, 5) = .Offset(9, 5) + s
                        For t = 0 To 1
                           .Offset(10, 5) = .Offset(10, 5) + t
                           For u = 0 To 1
                              .Offset(11, 5) = .Offset(11, 5) + u
                              For v = 0 To 1
                                 .Offset(12, 5) = .Offset(12, 5) + v
                                 If .Offset(15, 8) >= .Offset(17, 1) And .Offset(14, 6)_
                                    < Least Then
                                    For m = 1 To NumParts
                                       BackUp(m) = .Offset(m + 2, 5)
                                    Next
                                    Least = .Offset(14, 6)
                                    Replace = True
                                 End If
                                 .Offset(12, 5) = .Offset(12, 5) - v
                              Next
                              .Offset(11, 5) = .Offset(11, 5) - u
                           Next
                           .Offset(10, 5) = .Offset(10, 5) - t
                        Next
                        .Offset(9, 5) = .Offset(9, 5) - s
                     Next
                     .Offset(8, 5) = .Offset(8, 5) - r
                  Next
                  .Offset(7, 5) = .Offset(7, 5) - e
               Next
               .Offset(6, 5) = .Offset(6, 5) - d
            Next
            .Offset(5, 5) = .Offset(5, 5) - c
```

```
          Next
          .Offset(4, 5) = .Offset(4, 5) - b
        Next
        .Offset(3, 5) = .Offset(3, 5) - a
      Next
    Else
      For a = 0 To 1
        .Offset(3, 5) = .Offset(3, 5) + a
        For b = 0 To 1
          .Offset(4, 5) = .Offset(4, 5) + b
          For c = 0 To 1
            .Offset(5, 5) = .Offset(5, 5) + c
            For d = 0 To 1
              .Offset(6, 5) = .Offset(6, 5) + d
              For e = 0 To 1
                .Offset(7, 5) = .Offset(7, 5) + e
                If .Offset(15, 8) >= .Offset(17, 1) And .Offset(14, 6) < Least Then
                  For m = 1 To NumParts
                    BackUp(m) = .Offset(m + 2, 5)
                  Next
                  Least = .Offset(14, 6)
                  Replace = True
                End If
                .Offset(7, 5) = .Offset(7, 5) - e
              Next
              .Offset(6, 5) = .Offset(6, 5) - d
            Next
            .Offset(5, 5) = .Offset(5, 5) - c
          Next
          .Offset(4, 5) = .Offset(4, 5) - b
        Next
        .Offset(3, 5) = .Offset(3, 5) - a
      Next
    End If
    For i = 1 To NumParts
      .Offset(i + 2, 5) = .Offset(i + 2, 5) + 1
    Next
    For i = 1 To NumParts
      .Offset(i + 2, 5) = .Offset(i + 2, 5) + 1
      For j = 1 To NumParts
        If i <> j Then
          .Offset(j + 2, 5) = .Offset(j + 2, 5) - 1
          If .Offset(15, 8) >= .Offset(17, 1) And .Offset(14, 6) < Least Then
            For m = 1 To NumParts
              BackUp(m) = .Offset(m + 2, 5)
            Next
```

```
                Least = .Offset(14, 6)
                Replace = True
            End If
            .Offset(j + 2, 5) = .Offset(j + 2, 5) + 1
        End If
    Next
    .Offset(i + 2, 5) = .Offset(i + 2, 5) - 1
Next

    For i = 1 To NumParts
        .Offset(i + 2, 5) = BackUp(i)
    Next
    Cost = .Offset(14, 6)
End With

End Sub


Sub SaveSolution()
With Worksheets("Calc Sheet").Range("A1")
'Now save the solution (all Qp's and total cost) in the comparison array
        For i = 1 To NumParts
            Comparison(i, n - Range1 + 1) = .Offset(i + 2, 5)
        Next
        Comparison(NumParts + 1, n - Range1 + 1) = Cost
        Comparison(NumParts + 2, n - Range1 + 1) = n
        Comparison(NumParts + 3, n - Range1 + 1) = .Offset(18, 1)
        Worksheets("Qd Curves").Range("A1").Offset(n - Range1 + 1,_
        (2 * Problem) - 1) = n
        Worksheets("Qd Curves").Range("A1").Offset(n - Range1 + 1, (Problem)) = Cost
End With
End Sub


Sub TrendCheck()
'Update the Qd array and see if the last five solutions have been increasing
'If so, stop incrementing Qd
        j = 0
        For i = 3 To 2 Step -1
            Qd(i) = Qd(i - 1)
        Next
        Qd(1) = Cost
        MovingAvg2 = MovingAvg1
        MovingAvg1 = (Qd(1) + Qd(2) + Qd(3)) / 3
        If MovingAvg1 > MovingAvg2 Then
            Trend = Trend + 1
```

```
        Else
            Trend = 0
        End If
        If Trend > 5 Or n > 150 Then
            AllDone = True
            Trend = 0
        End If
End Sub


Sub ChooseBest()
'Loop through comparison array to choose the best solution
        Best = 1
        For i = 2 To n - Range1 + 1
            If Comparison(NumParts + 1, i) < Comparison(NumParts + 1, Best) Then
                Best = i
            End If
        Next
End Sub


Sub WriteSolution()
'Write best solution to Results worksheet
    With Worksheets("Results").Range("A1")
        .Offset(Problem, 0) = Problem
        For i = 1 To NumParts
            .Offset(Problem, i) = Comparison(i, Best)
        Next
        .Offset(Problem, 11) = Comparison(NumParts + 1, Best)
        .Offset(Problem, 12) = Comparison(NumParts + 2, Best)
        .Offset(Problem, 13) = Comparison(NumParts + 3, Best)
         EndTime = Timer
        .Offset(Problem, 14) = EndTime - StartTime
    End With
End Sub
```

APPENDIX D – Detailed Results of Runs

Table A1-1: ReNet Results, Single-Period Problem

St. Dev.	Problem	Part Order Quantities 1	2	3	4	5	6	7	8	9	10	Cost	Qd	SL	Time
	N-B-HH-10	1	4	5	7	10	12	14	16	18	20	6,014	59	0.968	0.61
	N-B-HL-10	0	4	7	11	15	20	25	29	32	35	16,021	119	0.976	0.61
	N-A-HH-10	1	4	5	7	10	12	14	16	18	20	5,146	59	0.968	0.50
	N-A-HL-10	0	0	0	0	0	0	5	11	18	23	25,990	208	0.953	0.52
	W-B-HH-10	1	7	11	16	21	24	29	34	38	43	8,492	59	0.965	0.52
	W-B-HL-10	0	2	7	12	18	22	27	32	37	42	9,877	65	0.986	0.50
	W-A-HH-10	1	7	11	16	21	24	29	34	38	43	5,569	59	0.965	0.50
0.01	W-A-HL-10	0	0	0	0	0	3	12	20	29	36	20,033	106	0.955	0.46
	N-B-HH-5	1	6	11	15	19						6,110	59	0.965	0.44
	N-B-HL-5	3	11	20	28	36						15,695	111	0.974	0.51
	N-A-HH-5	1	6	11	15	19						5,042	59	0.965	0.44
	N-A-HL-5	0	0	0	5	21	N/A					27,839	222	0.961	0.45
	W-B-HH-5	1	12	22	33	43						8,659	59	0.967	0.46
	W-B-HL-5	2	12	22	33	43						9,926	59	0.992	0.45
	W-A-HH-5	1	12	22	33	43						5,296	59	0.967	0.44
	W-A-HL-5	0	0	3	20	37						23,201	100	0.970	0.43
	N-B-HH-10	4	7	8	10	13	15	17	19	21	23	6,430	59	0.970	0.89
	N-B-HL-10	6	10	13	17	22	26	31	35	38	40	18,081	119	0.956	1.16
	N-A-HH-10	4	7	8	10	13	15	17	19	21	23	5,514	59	0.970	0.79
	N-A-HL-10	0	0	0	0	0	7	15	21	28	33	31,013	208	0.954	0.90
	W-B-HH-10	4	10	14	19	24	27	32	37	41	46	9,052	59	0.967	0.80
	W-B-HL-10	0	5	11	16	21	25	30	35	40	45	10,509	65	0.964	0.74
	W-A-HH-10	4	10	14	19	24	27	32	37	41	46	5,961	59	0.967	0.78
0.03	W-A-HL-10	0	0	0	0	2	8	17	25	34	41	22,025	106	0.958	0.69
	N-B-HH-5	4	9	13	18	22						6,507	59	0.971	0.58
	N-B-HL-5	8	17	25	33	40						17,383	111	0.950	0.76
	N-A-HH-5	4	9	13	18	22						5,396	59	0.971	0.58
	N-A-HL-5	0	0	0	14	30	N/A					32,503	222	0.953	0.60
	W-B-HH-5	4	15	25	35	46						9,196	59	0.970	0.59
	W-B-HL-5	4	15	25	35	45						10,436	59	0.958	0.57
	W-A-HH-5	4	15	25	35	46						5,674	59	0.970	0.58
	W-A-HL-5	0	0	7	24	41						25,073	100	0.953	0.52
	N-B-HH-10	6	10	11	13	16	18	20	22	24	26	6,828	59	0.954	1.16
	N-B-HL-10	12	16	19	23	28	32	37	41	44	46	20,147	119	0.953	1.70
	N-A-HH-10	6	10	11	13	16	18	20	22	24	26	5,825	59	0.954	1.02
	N-A-HL-10	0	0	0	1	9	18	26	32	38	43	36,358	208	0.950	1.40
	W-B-HH-10	6	13	17	22	27	30	35	40	45	49	9,603	59	0.954	1.04
	W-B-HL-10	4	9	14	19	24	28	33	39	44	47	11,207	65	0.951	1.03
	W-A-HH-10	6	13	17	22	27	30	35	40	45	49	6,295	59	0.954	1.02
0.05	W-A-HL-10	0	0	0	0	7	13	22	30	39	46	24,064	106	0.952	0.92
	N-B-HH-5	6	12	16	20	25						6,871	59	0.955	0.68
	N-B-HL-5	13	22	30	38	46						19,150	111	0.955	1.00
	N-A-HH-5	6	12	16	20	25						5,662	59	0.955	0.69
	N-A-HL-5	0	0	7	24	41	N/A					38,490	222	0.953	0.81
	W-B-HH-5	6	18	28	38	48						9,691	59	0.955	0.70
	W-B-HL-5	7	18	28	38	47						11,029	59	0.952	0.70
	W-A-HH-5	6	18	28	38	48						5,951	59	0.955	0.69
	W-A-HL-5	0	0	11	29	45						26,995	100	0.953	0.64

Table A1-2: SIPR Results, Single-Period Problem

St. Dev.	Problem	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)
		1	2	3	4	5	6	7	8	9	10	Disassembly	Purchase	Holding	Total			
0.01	N-B-HH-10	0	0	2	4	6	9	12	13	15	17	5,040	895	0.45	5,935	63	0.952	39.4
	N-B-HL-10	0	0	4	8	13	18	23	27	31	34	10,080	5,818	0.82	15,899	126	0.951	298.4
	N-A-HH-10	1	3	5	7	9	12	14	16	18	20	4,720	377	0.31	5,097	59	0.953	42.3
	N-A-HL-10	0	0	0	0	0	6	13	18	22	27	14,080	11,611	0.96	25,692	176	0.950	581.7
	W-B-HH-10	0	5	10	14	19	23	28	33	38	43	4,880	3,553	0.41	8,433	61	0.954	45.0
	W-B-HL-10	0	0	5	10	16	20	25	31	36	42	5,440	4,382	0.54	9,823	68	0.958	123.3
	W-A-HH-10	1	6	11	16	21	24	29	34	39	44	4,720	802	0.32	5,522	59	0.963	45.2
	W-A-HL-10	0	0	0	0	1	6	14	22	29	37	8,000	11,874	0.59	19,875	100	0.951	196.3
	N-B-HH-5	0	4	9	14	18						4,880	1,191	0.39	6,072	61	0.972	12.3
	N-B-HL-5	0	8	17	26	35						9,520	6,013	0.67	15,534	119	0.956	77.3
	N-A-HH-5	0	6	10	15	19						4,800	214	0.29	5,014	60	0.951	11.3
	N-A-HL-5	0	0	0	7	21	N/A					17,440	10,178	1.28	27,619	218	0.953	2.6
	W-B-HH-5	0	10	21	32	43						4,880	3,741	0.39	8,621	61	0.963	12.1
	W-B-HL-5	0	9	20	31	42						5,040	4,790	0.42	9,830	63	0.964	32.0
	W-A-HH-5	0	12	22	33	43						4,800	475	0.29	5,275	60	0.951	11.8
	W-A-HL-5	0	0	3	20	37						7,920	15,201	0.56	23,121	99	0.950	2.7
0.03	N-B-HH-10	0	1	3	5	7	10	13	15	17	19	5,280	1,044	1.20	6,326	66	0.952	101.9
	N-B-HL-10	50	50	50	50	50	50	50	50	50	50	80	17,200	0.00	17,280	1	1.000	9.1
	N-A-HH-10	1	3	6	9	11	14	17	19	21	23	4,960	420	0.96	5,381	62	0.952	116.8
	N-A-HL-10	50	50	50	50	50	50	50	50	50	50	80	26,982	0.00	27,062	1	1.000	9.3
	W-B-HH-10	0	5	10	15	21	24	30	35	40	46	5,200	3,767	1.17	8,968	65	0.951	115.6
	W-B-HL-10	1	6	11	16	21	25	30	35	40	45	5,120	5,342	1.15	10,463	64	0.952	234.5
	W-A-HH-10	0	6	12	17	22	26	32	37	43	48	5,040	778	0.99	5,819	63	0.954	124.8
	W-A-HL-10	50	50	50	50	50	50	50	50	50	50	80	20,810	0.01	20,890	1	1.000	434.6
	N-B-HH-5	0	4	9	14	19						5,200	1,216	1.09	6,417	65	0.958	26.7
	N-B-HL-5	50	50	50	50	50						80	16,934	0.00	17,014	1	1.000	4.4
	N-A-HH-5	0	6	11	17	22						5,040	231	0.90	5,271	63	0.961	29.6
	N-A-HL-5	50	50	50	50	50	N/A					80	28,804	0.00	28,884	1	1.000	2.8
	W-B-HH-5	0	11	23	34	45						5,120	3,993	1.06	9,114	64	0.956	27.8
	W-B-HL-5	2	13	23	34	45						4,960	5,435	1.01	10,396	62	0.952	58.6
	W-A-HH-5	0	12	23	35	47						5,040	493	0.87	5,534	63	0.951	30.2
	W-A-HL-5	50	50	50	50	50						80	23,940	0.01	24,020	1	1.000	2.9
0.05	N-B-HH-10	0	2	4	7	9	12	15	17	19	21	5,520	1,251	2.05	6,773	69	0.950	192.7
	N-B-HL-10	50	50	50	50	50	50	50	50	50	50	80	17,200	0.00	17,280	1	0.998	10.3
	N-A-HH-10	0	2	6	8	11	15	18	21	23	25	5,360	320	1.71	5,682	67	0.950	211.2
	N-A-HL-10	50	50	50	50	50	50	50	50	50	50	80	26,982	0.00	27,062	1	0.998	9.7
	W-B-HH-10	14	18	22	26	30	33	37	41	45	49	4,000	5,531	1.50	9,533	50	0.954	64.9
	W-B-HL-10	50	50	50	50	50	50	50	50	50	50	80	10,681	0.10	10,761	1	0.999	10.1
	W-A-HH-10	0	5	13	19	25	29	36	41	48	54	5,360	789	1.72	6,150	67	0.951	225.7
	W-A-HL-10	50	50	50	50	50	50	50	50	50	50	80	20,810	0.01	20,890	1	0.999	10.3
	N-B-HH-5	0	4	10	15	20						5,520	1,295	1.87	6,817	69	0.952	47.7
	N-B-HL-5	50	50	50	50	50						80	16,934	0.01	17,014	1	0.999	3.0
	N-A-HH-5	0	7	12	19	24						5,280	257	1.52	5,539	66	0.953	52.3
	N-A-HL-5	50	50	50	50	50	N/A					80	28,804	0.00	28,884	1	0.999	3.1
	W-B-HH-5	9	19	29	39	49						4,400	5,248	1.49	9,649	55	0.950	46.6
	W-B-HL-5	50	50	50	50	50						80	10,700	0.10	10,780	1	0.999	4.1
	W-A-HH-5	0	14	25	39	51						5,280	550	1.53	5,831	66	0.955	54.2
	W-A-HL-5	50	50	50	50	50						80	23,940	0.01	24,020	1	0.999	3.2

Table A1-3: Sensitivity Results for Core Purchase and Disassembly Costs

Problem	Core Cost	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)
		1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Total			
N-B-HH-10	10	0	0	0	0	0	0	0	0	0	3	24.00	0.56	960.00	984.56	96	0.953	190
	20	0	0	0	0	0	0	0	3	5	9	155.00	0.81	1,740.00	1,895.81	87	0.953	184
	30	0	0	0	0	0	0	3	6	8	11	273.00	0.97	2,460.00	2,733.97	82	0.953	179
	40	0	0	0	0	0	3	6	9	11	14	441.00	1.02	3,080.00	3,522.02	77	0.955	174
	50	0	0	0	1	4	7	10	12	14	17	711.00	0.93	3,550.00	4,261.93	71	0.953	168
	60	0	0	0	1	4	7	10	12	14	17	711.00	1.11	4,260.00	4,972.11	71	0.953	164
	70	0	0	1	3	6	9	12	14	16	19	900.00	1.13	4,760.00	5,661.13	68	0.953	158
	80	0	1	3	5	7	10	13	15	17	19	1,044.46	1.20	5,280.00	6,325.66	66	0.952	157
	90	50	50	50	50	50	50	50	50	50	50	6,923.00	-	-	6,923.00	0	1.000	12
	100	50	50	50	50	50	50	50	50	50	50	6,923.00	-	-	6,923.00	0	1.000	14
	110	50	50	50	50	50	50	50	50	50	50	6,923.00	-	-	6,923.00	0	1.000	8
N-A-HH-10	10	0	0	0	1	3	8	11	13	15	17	65.99	0.25	710.00	776.24	71	0.954	188
	20	0	0	2	4	6	10	13	15	17	20	128.00	0.34	1,340.00	1,468.34	67	0.953	181
	30	0	0	2	5	8	11	14	16	18	20	150.80	0.45	1,980.00	2,131.25	66	0.951	179
	40	0	0	2	5	8	11	14	16	18	20	150.80	0.60	2,640.00	2,791.40	66	0.951	177
	50	0	0	2	5	8	11	14	16	18	20	150.80	0.75	3,300.00	3,451.55	66	0.951	175
	60	0	2	4	6	9	13	15	17	19	21	268.07	0.79	3,840.00	4,108.86	64	0.956	174
	70	0	2	4	6	9	13	15	17	19	21	268.07	0.92	4,480.00	4,748.99	64	0.956	168
	80	1	3	6	9	11	14	17	19	21	23	419.61	0.96	4,960.00	5,380.57	62	0.952	170
	90	1	3	6	9	11	14	17	19	21	23	419.61	1.08	5,580.00	6,000.69	62	0.952	167
	100	50	50	50	50	50	50	50	50	50	50	6,113.50	-	-	6,113.50	0	1.000	10
	110	50	50	50	50	50	50	50	50	50	50	6,113.50	-	-	6,113.50	0	1.000	10
W-B-HH-10	10	0	0	0	0	0	0	0	11	25	37	1,069.00	1.15	1,570.00	2,640.15	157	0.951	402
	20	0	0	0	0	0	4	13	22	32	41	1,699.00	1.20	2,300.00	4,000.20	115	0.950	325
	30	0	0	0	0	6	12	19	27	35	43	2,212.00	1.16	2,880.00	5,093.16	96	0.950	264
	40	0	0	0	5	12	17	24	31	37	44	2,709.00	1.05	3,320.00	6,030.05	83	0.951	243
	50	0	0	0	7	13	18	24	31	38	44	2,801.00	1.22	4,050.00	6,852.22	81	0.950	204
	60	0	0	6	12	17	22	27	33	39	45	3,287.00	1.08	4,320.00	7,608.08	72	0.951	183
	70	0	4	9	15	20	24	30	35	40	45	3,690.56	1.05	4,620.00	8,311.61	66	0.950	162
	80	0	5	10	15	21	24	30	35	40	46	3,766.95	1.17	5,200.00	8,968.12	65	0.951	158
	90	50	50	50	50	50	50	50	50	50	50	9,319.50	-	-	9,319.50	0	1.000	8
	100	50	50	50	50	50	50	50	50	50	50	9,319.50	-	-	9,319.50	0	1.000	8
	110	50	50	50	50	50	50	50	50	50	50	9,319.50	-	-	9,319.50	0	1.000	8
W-A-HH-10	10	0	0	0	6	13	17	25	33	40	47	239.20	0.40	810.00	1,049.60	81	0.950	238
	20	0	0	6	12	17	23	29	34	41	47	375.14	0.42	1,420.00	1,795.56	71	0.951	204
	30	0	0	7	13	19	23	29	35	42	48	400.18	0.59	2,100.00	2,500.77	70	0.953	204
	40	0	0	7	13	19	23	29	35	42	48	400.18	0.78	2,800.00	3,200.96	70	0.953	201
	50	0	2	9	15	20	24	31	36	43	48	534.30	0.79	3,350.00	3,885.09	67	0.950	187
	60	0	4	10	15	21	25	31	36	42	48	645.34	0.83	3,900.00	4,546.17	65	0.957	182
	70	0	5	11	16	21	25	31	36	43	48	706.32	0.91	4,480.00	5,187.23	64	0.957	180
	80	0	6	12	17	22	26	32	37	43	48	778.46	0.99	5,040.00	5,819.45	63	0.954	175
	90	1	6	13	18	23	27	32	37	43	48	859.96	1.08	5,580.00	6,441.04	62	0.951	173
	100	50	50	50	50	50	50	50	50	50	50	6,519.00	-	-	6,519.00	0	1.000	10
	110	50	50	50	50	50	50	50	50	50	50	6,519.00	-	-	6,519.00	0	1.000	8

Table A1-4: Sensitivity Results for Target Service Level

Problem	TSL	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)
		1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Total			
N-B-HH-10	0.5	0	0	2	4	7	9	12	14	16	18	939.00	0.71	5,120	6,059.71	64	0.508	139
	0.55	0	0	2	4	7	10	12	14	16	19	961.00	0.74	5,120	6,081.74	64	0.561	136
	0.6	0	1	2	4	7	10	12	14	16	19	978.46	0.77	5,120	6,099.23	64	0.600	142
	0.65	0	0	1	4	6	10	12	14	16	18	921.00	0.84	5,200	6,121.84	65	0.652	147
	0.7	0	0	2	4	6	10	12	14	16	18	938.00	0.86	5,200	6,138.86	65	0.704	152
	0.75	0	0	2	4	7	10	12	14	16	19	961.00	0.90	5,200	6,161.90	65	0.757	160
	0.80	0	0	1	4	6	9	12	14	16	18	907.00	0.98	5,280	6,187.98	66	0.801	166
	0.85	0	0	2	4	7	9	12	14	16	18	939.00	1.03	5,280	6,220.03	66	0.856	174
	0.90	0	1	3	5	8	10	13	15	17	19	1,059.46	1.06	5,200	6,260.52	65	0.906	187
	0.95	0	1	3	5	7	10	13	15	17	19	1,044.46	1.20	5,280	6,325.66	66	0.952	205
	0.96	0	1	3	5	8	10	13	15	17	19	1,059.46	1.23	5,280	6,340.69	66	0.960	209
	0.97	0	1	3	5	8	11	13	15	18	20	1,091.46	1.28	5,280	6,372.74	66	0.972	217
	0.98	0	1	3	5	7	10	13	15	17	19	1,044.46	1.37	5,360	6,405.83	67	0.981	224
	0.99	0	2	4	6	8	12	14	16	18	20	1,178.92	1.43	5,280	6,460.35	66	0.990	240
N-A-HH-10	0.5	0	1	5	7	9	13	16	18	20	22	242.34	0.47	4,880	5,122.81	61	0.518	176
	0.55	0	1	5	8	10	14	16	18	20	22	256.34	0.50	4,880	5,136.84	61	0.557	170
	0.6	0	3	6	9	11	15	17	19	21	23	364.61	0.52	4,800	5,165.13	60	0.602	173
	0.65	0	2	5	7	10	14	16	18	20	22	292.34	0.55	4,880	5,172.89	61	0.660	178
	0.7	0	2	6	8	10	14	17	19	20	22	307.64	0.58	4,880	5,188.22	61	0.701	179
	0.75	0	3	6	8	10	14	16	18	20	22	348.34	0.65	4,880	5,228.99	61	0.767	184
	0.80	0	2	5	7	9	13	16	18	20	22	284.34	0.71	4,960	5,245.05	62	0.816	191
	0.85	0	2	6	8	10	14	16	18	20	22	306.34	0.75	4,960	5,267.09	62	0.854	197
	0.90	0	2	4	7	9	13	16	18	20	21	275.87	0.88	5,040	5,316.75	63	0.905	204
	0.95	1	3	6	9	11	14	17	19	21	23	419.61	0.96	4,960	5,380.57	62	0.952	223
	0.96	0	2	4	7	9	13	16	18	20	22	276.34	1.06	5,120	5,397.40	64	0.964	231
	0.97	0	2	5	7	9	13	16	18	20	22	284.34	1.08	5,120	5,405.42	64	0.971	236
	0.98	0	1	5	7	9	13	16	18	20	22	242.34	1.19	5,200	5,443.53	65	0.980	248
	0.99	0	2	5	7	9	13	16	18	20	22	284.34	1.27	5,200	5,485.61	65	0.990	269
W-B-HH-10	0.5	0	3	9	14	19	23	28	33	38	43	3,487.17	0.71	5,120	8,607.88	64	0.508	144
	0.55	0	4	9	14	20	23	29	34	39	44	3,590.56	0.72	5,040	8,631.28	63	0.557	139
	0.6	0	4	9	14	19	23	28	33	39	44	3,538.56	0.77	5,120	8,659.33	64	0.618	147
	0.65	0	4	9	15	21	24	29	34	39	44	3,647.56	0.79	5,040	8,688.35	63	0.650	152
	0.7	0	4	9	14	19	24	29	34	39	44	3,589.56	0.84	5,120	8,710.40	64	0.705	157
	0.75	0	5	10	15	20	24	29	34	40	45	3,700.95	0.85	5,040	8,741.80	63	0.758	163
	0.80	0	4	10	15	20	24	29	34	39	45	3,663.56	0.92	5,120	8,784.48	64	0.815	168
	0.85	0	5	10	15	20	24	29	34	40	45	3,700.95	0.97	5,120	8,821.92	64	0.858	178
	0.90	0	5	10	15	21	24	30	35	40	45	3,752.95	1.03	5,120	8,873.98	64	0.900	188
	0.95	0	5	10	15	21	24	30	35	40	46	3,766.95	1.17	5,200	8,968.12	65	0.951	209
	0.96	0	6	11	16	21	25	30	36	41	46	3,879.34	1.19	5,120	9,000.53	64	0.960	214
	0.97	0	5	10	16	21	25	30	35	41	47	3,833.95	1.26	5,200	9,035.21	65	0.970	221
	0.98	0	6	11	16	21	25	30	36	41	46	3,879.34	1.31	5,200	9,080.65	65	0.981	237
	0.99	0	5	11	16	21	25	30	36	41	47	3,870.95	1.43	5,280	9,152.38	66	0.990	264
W-A-HH-10	0.5	0	4	11	16	21	25	31	36	42	47	659.24	0.47	4,880	5,539.71	61	0.508	182
	0.55	0	4	12	17	22	25	31	36	42	47	678.24	0.50	4,880	5,558.74	61	0.556	178
	0.6	0	6	13	18	22	26	32	37	43	48	792.46	0.52	4,800	5,592.98	60	0.606	181
	0.65	0	5	11	17	22	25	31	36	42	47	717.08	0.55	4,880	5,597.63	61	0.657	183
	0.7	0	5	12	17	23	26	32	37	42	48	736.48	0.59	4,880	5,617.07	61	0.702	187
	0.75	0	6	12	17	22	26	32	37	42	48	778.32	0.65	4,880	5,658.97	61	0.769	191
	0.80	0	5	11	16	22	25	31	36	42	48	711.18	0.70	4,960	5,671.88	62	0.808	201
	0.85	0	4	11	16	21	26	31	36	42	48	663.84	0.80	5,040	5,704.64	63	0.851	206
	0.90	0	5	11	16	21	25	31	36	42	48	706.18	0.87	5,040	5,747.05	63	0.902	216
	0.95	0	6	12	17	22	26	32	37	43	48	778.46	0.99	5,040	5,819.45	63	0.954	232
	0.96	0	5	11	16	22	25	31	37	43	48	712.12	1.05	5,120	5,833.17	64	0.962	240
	0.97	0	5	12	17	22	26	32	37	43	48	731.62	1.09	5,120	5,852.71	64	0.973	246
	0.98	1	6	12	18	23	27	32	38	43	49	852.86	1.12	5,040	5,893.98	63	0.980	251
	0.99	0	5	12	17	22	26	32	37	44	49	731.86	1.26	5,200	5,933.12	65	0.990	279

Table A1-5: Sensitivity Results for End-Item Demand Level

Problem	Demand	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)	Unit Cost	DA Ratio
		1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Total					
N-B-HH-10	2	0	0	0	0	0	1	1	1	1	1	56.00	0.13	240	296.13	3	0.961	4	148.06	1.50
	4	0	0	0	0	0	1	1	1	1	2	64.00	0.19	480	544.19	6	0.958	8	136.05	1.50
	6	0	0	1	1	1	1	2	2	2	3	153.00	0.21	640	793.21	8	0.952	11	132.20	1.33
	8	0	0	0	1	1	2	2	2	3	3	160.00	0.25	880	1,040.25	11	0.961	16	130.03	1.38
	10	0	1	1	1	2	2	3	3	4	4	251.46	0.28	1,040	1,291.74	13	0.958	20	129.17	1.30
	20	0	0	0	1	2	3	4	5	6	7	310.00	0.57	2,240	2,550.57	28	0.955	53	127.53	1.40
	30	0	0	1	3	4	6	8	9	10	12	605.00	0.75	3,200	3,805.75	40	0.952	95	126.86	1.33
	40	0	1	2	4	6	8	10	12	13	15	825.46	0.98	4,240	5,066.44	53	0.953	146	126.66	1.33
	50	0	1	3	5	7	10	13	15	17	19	1,044.46	1.20	5,280	6,325.66	66	0.952	209	126.51	1.32
	60	0	2	4	7	10	13	16	18	21	24	1,347.92	1.40	6,240	7,589.32	78	0.951	283	126.49	1.30
	70	0	1	4	7	11	15	18	21	24	27	1,486.46	1.66	7,360	8,848.12	92	0.950	363	126.40	1.31
	80	0	2	5	8	12	17	21	24	27	31	1,712.92	1.89	8,400	10,114.81	105	0.951	462	126.44	1.31
	90	0	2	6	9	14	19	23	27	31	35	1,933.92	2.12	9,440	11,376.04	118	0.951	560	126.40	1.31
	100	0	1	5	10	15	20	26	30	34	39	2,077.46	2.39	10,560	12,639.85	132	0.950	686	126.40	1.32
N-A-HH-10	2	0	0	0	0	0	1	1	1	1	1	3.27	0.11	240	243.38	3	0.961	4	121.69	1.50
	4	1	1	2	2	2	2	2	2	2	3	147.01	0.07	320	467.08	4	0.986	8	116.77	1.00
	6	0	0	1	1	1	1	2	2	2	3	27.01	0.19	640	667.20	8	0.952	12	111.20	1.33
	8	0	1	1	2	2	3	3	3	4	4	86.78	0.20	800	886.98	10	0.951	16	110.87	1.25
	10	0	0	1	2	2	3	3	4	4	5	45.85	0.23	1,040	1,086.08	13	0.950	22	108.61	1.30
	20	1	2	3	4	5	6	7	8	9	9	247.43	0.38	1,920	2,167.81	24	0.957	55	108.39	1.20
	30	1	2	4	5	6	8	10	11	12	14	278.18	0.59	2,960	3,238.77	37	0.952	102	107.96	1.23
	40	1	2	5	6	8	11	13	15	16	18	317.56	0.81	4,000	4,318.37	50	0.957	160	107.96	1.25
	50	1	3	6	9	11	14	17	19	21	23	419.61	0.96	4,960	5,380.57	62	0.952	228	107.61	1.24
	60	0	3	6	8	11	16	19	21	24	26	365.12	1.19	6,080	6,446.31	76	0.951	310	107.44	1.27
	70	1	3	8	10	14	19	23	26	28	31	483.27	1.37	7,040	7,524.64	88	0.951	406	107.49	1.26
	80	0	4	8	12	16	21	26	29	32	35	505.05	1.56	8,080	8,586.61	101	0.951	509	107.33	1.26
	90	0	4	9	13	17	24	29	32	36	39	536.83	1.76	9,120	9,658.59	114	0.953	629	107.32	1.27
	100	0	4	9	14	19	26	32	36	40	43	567.21	1.97	10,160	10,729.18	127	0.951	764	107.29	1.27
W-B-HH-10	2	0	0	1	1	1	1	1	2	2	2	185.00	0.14	240	425.14	3	0.970	6	212.57	1.50
	4	0	0	1	1	2	2	2	3	3	4	298.00	0.19	480	778.19	6	0.953	11	194.55	1.50
	6	0	1	1	2	3	3	4	4	5	6	485.39	0.21	640	1,125.60	8	0.961	15	187.60	1.33
	8	0	0	0	1	2	3	4	5	6	7	448.00	0.32	1,040	1,488.32	13	0.952	22	186.04	1.63
	10	0	1	2	3	4	5	6	7	8	10	764.39	0.25	1,040	1,804.64	13	0.952	26	180.46	1.30
	20	0	2	4	6	8	10	12	14	16	19	1,514.78	0.48	2,080	3,595.26	26	0.951	57	179.76	1.30
	30	0	3	6	9	12	15	18	21	24	28	2,265.17	0.71	3,120	5,385.88	39	0.951	98	179.53	1.30
	40	0	4	8	12	16	20	24	28	32	37	3,015.56	0.94	4,160	7,176.50	52	0.951	150	179.41	1.30
	50	0	5	10	15	21	24	30	35	40	46	3,766.95	1.17	5,200	8,968.12	65	0.951	212	179.36	1.30
	60	0	6	12	18	25	29	36	42	48	55	4,517.34	1.40	6,240	10,758.74	78	0.952	286	179.31	1.30
	70	0	7	14	21	29	34	42	49	56	64	5,267.73	1.64	7,280	12,549.37	91	0.952	372	179.28	1.30
	80	0	8	16	24	33	39	47	56	65	73	6,016.12	1.86	8,320	14,337.98	104	0.951	468	179.22	1.30
	90	0	9	18	27	37	44	53	63	73	82	6,766.51	2.10	9,360	16,128.61	117	0.951	567	179.21	1.30
	100	0	10	21	31	42	49	60	70	81	91	7,593.90	2.30	10,320	17,916.20	129	0.950	681	179.16	1.29
W-A-HH-10	2	0	0	1	1	1	1	1	2	2	2	26.58	0.11	240	266.69	3	0.970	6	133.34	1.50
	4	0	1	1	2	2	3	3	3	4	4	96.70	0.12	400	496.82	5	0.951	11	124.21	1.25
	6	1	1	2	3	3	4	4	5	5	6	181.14	0.17	560	741.31	7	0.966	16	123.55	1.17
	8	1	2	3	3	4	5	6	6	7	8	248.76	0.20	720	968.96	9	0.953	21	121.12	1.13
	10	0	1	2	3	4	5	6	7	8	10	137.06	0.24	1,040	1,177.30	13	0.952	27	117.73	1.30
	20	1	3	6	8	10	11	13	15	17	19	423.30	0.38	1,920	2,343.68	24	0.952	65	117.18	1.20
	30	1	4	7	11	14	16	19	22	26	29	552.50	0.60	2,960	3,513.10	37	0.959	108	117.10	1.23
	40	0	4	9	13	17	20	25	29	34	38	569.12	0.80	4,080	4,649.92	51	0.953	167	116.25	1.28
	50	0	6	12	17	22	26	32	37	43	48	778.46	0.99	5,040	5,819.45	63	0.954	238	116.39	1.26
	60	0	5	13	20	26	31	38	44	51	58	813.84	1.23	6,160	6,975.07	77	0.951	322	116.25	1.28
	70	0	7	16	23	30	36	44	51	60	67	1,005.78	1.38	7,120	8,127.16	89	0.952	419	116.10	1.27
	80	0	9	19	27	35	41	51	59	69	77	1,210.62	1.56	8,080	9,292.18	101	0.951	532	116.15	1.26
	90	0	8	20	30	39	46	57	66	77	87	1,246.00	1.79	9,200	10,447.79	115	0.950	646	116.09	1.28
	100	0	10	23	33	44	51	63	74	86	96	1,443.74	1.96	10,160	11,605.70	127	0.951	780	116.06	1.27

Table A1-6: Sensitivity Results for Standard Deviation of Yield Distribution

Problem	St. Dev.	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)
		1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Cost			
1	0.001	0	1	3	6	9	12	15	18	20	23	1,681.36	0.18	6,400	8,082	80	0.988	99
1	0.01	0	1	3	6	9	12	16	18	21	23	1,711.36	0.57	6,640	8,352	83	0.955	161
1	0.02	0	1	4	7	10	14	17	20	22	25	1,894.92	1.04	6,800	8,696	85	0.950	254
1	0.03	0	3	5	8	11	15	19	21	24	27	2,126.70	1.56	6,960	9,088	87	0.952	361
1	0.04	2	5	7	10	14	17	21	24	26	29	2,541.26	2.06	6,960	9,503	87	0.950	462
1	0.05	50	50	50	50	50	50	50	50	50	50	9,539.00	-	-	9,539	0	1.000	18
1	0.06	50	50	50	50	50	50	50	50	50	50	9,539.00	-	-	9,539	0	1.000	18
1	0.07	50	50	50	50	50	50	50	50	50	50	9,539.00	-	-	9,539	0	1.000	18
1	0.08	50	50	50	50	50	50	50	50	50	50	9,539.00	-	-	9,539	0	1.000	18
1	0.09	50	50	50	50	50	50	50	50	50	50	9,539.00	-	-	9,539	0	1.000	18
1	0.1	50	50	50	50	50	50	50	50	50	50	9,539.00	-	-	9,539	0	1.000	18
1	0.11	50	50	50	50	50	50	50	50	50	50	9,539.00	-	-	9,539	0	1.000	18
2	0.001	0	2	5	7	10	13	16	19	21	23	416.46	0.14	6,240	6,657	78	1.000	93
2	0.01	0	2	5	7	11	14	18	20	22	25	433.22	0.44	6,400	6,834	80	0.953	172
2	0.02	0	3	6	8	12	15	20	22	25	27	528.70	0.86	6,560	7,090	82	0.950	273
2	0.03	0	3	7	9	13	16	21	24	27	29	562.82	1.34	6,800	7,364	85	0.954	390
2	0.04	0	2	7	10	13	17	23	26	29	32	522.24	1.85	7,120	7,644	89	0.950	532
2	0.05	33	34	36	37	38	40	42	43	44	45	5,582.54	0.80	2,480	8,063	31	0.950	146
2	0.06	50	50	50	50	50	50	50	50	50	50	8,123.00	-	-	8,123	0	1.000	28
2	0.07	50	50	50	50	50	50	50	50	50	50	8,123.00	-	-	8,123	0	1.000	28
2	0.08	50	50	50	50	50	50	50	50	50	50	8,123.00	-	-	8,123	0	1.000	28
2	0.09	50	50	50	50	50	50	50	50	50	50	8,123.00	-	-	8,123	0	1.000	28
2	0.1	50	50	50	50	50	50	50	50	50	50	8,123.00	-	-	8,123	0	1.000	28
2	0.11	50	50	50	50	50	50	50	50	50	50	8,123.00	-	-	8,123	0	1.000	28

Table A1-7: Multi-Period Results, Increasing Demand Pattern

Pattern	Problem	Period	Part Order Quantities										Costs				Qd	Svc. Level	Time (s)	Excess by Period (units)									
			1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Total				1	2	3	4	5	6	7	8	9	10
Increasing	1	1	0	1	3	5	7	10	13	15	17	19	1,044.46	1.20	5,280	6,326	66	0.952	48.9	6	5	5	5	4	4	5	5	5	5
Increasing	1	2	0	2	4	7	10	13	15	17	20	22	1,297.92	0.28	5,680	6,978	71	0.951	13.4	1	0	0	1	1	1	1	1	1	1
Increasing	1	3	0	3	6	7	11	15	18	21	24	27	1,555.38	1.44	7,200	8,757	90	0.950	135.4	6	6	7	5	5	6	5	5	6	6
Increasing	1	4	0	3	4	9	13	16	21	24	26	30	1,711.38	0.51	7,680	9,392	96	0.950	36.3	2	2	1	2	2	1	2	2	2	3
Increasing	1	5	0	1	6	9	13	19	23	26	30	33	1,864.46	1.68	9,280	11,146	116	0.951	225.0	8	6	7	7	6	7	7	6	7	6
Increasing	2	1	1	3	6	9	11	14	17	19	21	23	419.61	0.96	4,960	5,381	62	0.952	65.4	3	3	4	6	5	6	6	6	6	7
Increasing	2	2	0	3	5	5	9	12	15	17	20	21	311.57	0.41	5,760	6,072	72	0.952	19.9	1	2	2	0	1	1	1	1	2	1
Increasing	2	3	0	1	6	10	13	18	22	25	26	30	316.50	1.07	7,040	7,358	88	0.951	141.7	4	3	5	6	6	7	8	8	7	8
Increasing	2	4	1	5	8	11	13	18	21	24	28	30	562.20	0.61	7,600	8,163	95	0.951	47.1	2	3	3	3	1	2	2	2	3	2
Increasing	2	5	0	2	7	12	17	23	28	31	34	38	427.06	1.20	8,960	9,388	112	0.950	229.4	5	4	5	7	7	9	9	9	8	9
Increasing	3	1	0	5	10	15	21	24	30	35	40	46	3,766.95	1.17	5,200	8,968	65	0.951	66.2	5	5	4	4	5	4	5	5	4	5
Increasing	3	2	0	6	12	18	23	28	33	39	45	51	4,261.34	0.33	5,680	9,942	71	0.952	16.6	1	1	1	1	1	1	1	1	1	1
Increasing	3	3	0	6	14	21	28	34	41	48	55	63	5,164.34	1.39	7,200	12,366	90	0.950	152.4	6	5	6	5	5	6	6	5	5	6
Increasing	3	4	0	8	15	24	31	36	44	53	61	68	5,674.12	0.55	7,680	13,355	96	0.951	40.2	2	2	1	2	2	1	2	3	3	2
Increasing	3	5	0	9	19	27	36	44	52	61	70	81	6,660.51	1.60	9,120	15,782	114	0.952	207.4	6	6	7	6	6	7	6	6	6	8
Increasing	4	1	0	6	12	17	22	26	32	37	43	48	778.46	0.99	5,040	5,819	63	0.954	56.3	3	4	5	5	5	5	6	6	7	7
Increasing	4	2	0	5	12	18	23	28	34	40	45	51	756.60	0.34	5,760	6,517	72	0.950	20.3	1	1	1	2	1	2	2	2	1	2
Increasing	4	3	0	8	16	22	31	35	43	50	59	66	1,045.08	1.10	6,960	8,006	87	0.951	139.1	4	5	6	5	7	5	6	6	9	9
Increasing	4	4	0	6	15	24	30	38	46	54	60	68	970.44	0.54	7,760	8,731	97	0.952	54.8	2	1	2	3	1	3	3	4	2	2
Increasing	4	5	0	10	19	28	39	44	54	62	75	84	1,303.90	1.38	8,960	10,265	112	0.953	231.4	5	6	6	6	8	6	7	6	10	10

Table A1-8: Multi-Period Results, Decreasing Demand Pattern

Problem	Period	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)	Excess by Period (units)									
		1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Total				1	2	3	4	5	6	7	8	9	10
1	1	0	2	6	9	14	19	23	27	31	35	1933.92	2.12	9440	11,376	118	0.951	250.4	10	8	9	8	8	9	8	8	9	9
1	2	0	0	1	4	8	12	16	19	22	26	1210	0.34	7920	9,130	99	0.964	7.5	4	1	0	0	0	0	0	0	0	1
1	3	0	4	8	11	14	17	21	24	26	28	1839.84	1.42	6960	8,801	87	0.950	118.3	4	5	6	7	6	6	6	7	6	5
1	4	0	0	1	3	6	9	12	14	17	21	926	0.25	5920	6,846	74	0.953	4.9	2	1	0	0	0	0	0	0	0	1
1	5	0	1	4	6	9	12	14	16	18	19	1168.46	1.08	5120	6,290	64	0.952	35.6	4	3	4	4	5	5	4	5	5	4
2	1	0	4	9	13	17	24	29	32	36	39	536.83	1.76	9120	9,659	114	0.953	295.4	6	7	9	9	9	11	11	11	12	11
2	2	0	3	6	10	13	16	20	23	26	28	394.96	0.18	7440	7,835	93	0.950	7.0	0	0	0	1	1	0	0	0	1	0
2	3	1	3	8	9	13	19	23	26	27	31	469.77	1.35	7040	7,511	88	0.951	155.0	5	5	7	5	6	8	9	9	8	9
2	4	0	2	4	7	9	12	15	17	19	21	273.07	0.14	5600	5,873	70	0.953	3.6	0	0	0	0	0	0	0	0	0	0
2	5	1	3	6	9	11	14	17	19	21	23	419.61	0.96	4960	5,381	62	0.952	51.0	3	3	4	6	5	6	6	6	6	7
3	1	0	9	18	27	37	44	53	63	73	82	6766.51	2.10	9360	16,129	117	0.951	270.8	9	9	8	8	9	8	8	9	9	9
3	2	0	2	10	18	26	32	41	48	56	65	4899.78	0.32	8000	12,900	100	0.951	8.1	5	0	0	0	0	0	1	0	0	1
3	3	1	12	19	26	32	37	43	51	58	64	5798.68	1.31	6640	12,440	83	0.950	107.2	1	5	6	6	5	6	5	6	7	6
3	4	0	2	7	14	21	24	31	37	42	49	3737.78	0.35	6080	9,818	76	0.953	11.3	4	1	0	1	1	0	1	1	0	1
3	5	0	7	13	17	22	27	30	35	41	45	3988.73	0.84	4800	8,790	60	0.951	27.4	1	3	4	3	3	5	3	3	4	4
4	1	0	8	20	30	39	46	57	66	77	87	1246	1.79	9200	10,448	115	0.950	304.6	7	6	9	10	9	10	11	11	13	14
4	2	0	8	15	23	30	36	43	51	58	67	1043.34	0.19	7440	8,484	93	0.959	6.9	0	0	0	0	0	0	0	1	0	1
4	3	0	7	16	23	30	36	44	50	60	66	1004.88	1.38	7120	8,126	89	0.952	160.2	5	5	7	7	7	7	8	7	10	9
4	4	1	6	12	17	23	27	33	38	44	50	848.1	0.13	5520	6,368	69	0.960	3.5	0	0	0	0	0	0	0	0	0	1
4	5	0	6	12	17	22	26	32	37	43	47	778.36	0.99	5040	5,819	63	0.954	50.1	3	4	5	5	5	5	6	6	7	6

Table A1-9: Multi-Period Results, Spike Demand Pattern

Period	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)	Excess by Period (units)									
	1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Total				1	2	3	4	5	6	7	8	9	10
1	0	1	3	5	7	10	13	15	17	19	1044.46	1.20	5280	6,326	66	0.952	41.8	6	5	5	5	4	4	5	5	5	5
2	0	3	6	8	13	17	19	22	25	28	1670.38	0.56	7200	8,871	90	0.950	37.3	1	2	2	1	3	3	2	2	2	2
3	0	2	6	11	14	19	25	29	33	38	2055.92	1.86	10320	12,378	129	0.950	279.9	9	7	7	9	6	6	7	7	7	8
4	0	2	5	8	12	16	18	21	24	26	1556.92	0.22	7040	8,597	88	0.951	8.3	0	0	0	0	1	1	0	0	1	0
5	0	1	3	5	6	9	13	15	16	19	1005.46	1.14	5280	6,287	66	0.952	46.7	6	5	5	5	3	3	5	5	4	5
1	1	3	6	9	11	14	17	19	21	23	419.61	0.96	4960	5,381	62	0.952	48.1	3	3	4	6	5	6	6	6	6	7
2	0	3	7	8	13	17	21	24	26	28	393.26	0.67	7280	7,674	91	0.951	50.1	2	2	3	2	3	3	4	4	3	3
3	0	3	8	13	17	24	29	33	37	41	488.87	1.46	10000	10,490	125	0.950	278.2	6	5	6	8	7	9	9	9	9	9
4	0	3	7	8	12	15	19	22	24	27	380.19	0.25	7040	7,420	88	0.954	9.5	0	1	1	0	1	0	1	1	1	1
5	1	2	5	9	10	14	16	18	20	22	360.34	0.85	4960	5,321	62	0.952	50.4	3	2	3	6	4	6	5	5	5	6
1	0	5	10	15	21	24	30	35	40	46	3766.95	1.17	5200	8,968	65	0.951	46.3	5	5	4	4	5	4	5	5	4	5
2	0	7	15	22	29	35	42	49	57	64	5341.73	0.63	7280	12,622	91	0.952	45.2	2	2	3	2	2	2	2	2	3	2
3	0	10	19	30	40	48	58	69	78	90	7366.9	1.78	10160	17,529	127	0.952	256.4	7	7	6	7	7	7	7	8	7	9
4	0	7	15	21	28	33	40	47	55	61	5128.73	0.22	7040	12,169	88	0.951	8.4	0	0	1	0	0	0	0	0	1	0
5	0	5	9	15	21	24	30	35	39	46	3730.95	1.13	5200	8,932	65	0.951	57.9	5	5	3	4	5	4	5	5	3	5
1	0	6	12	17	22	26	32	37	43	48	778.46	0.99	5040	5,819	63	0.954	48.1	3	4	5	5	5	5	6	6	7	7
2	0	7	15	22	30	35	43	50	58	65	985	0.61	7280	8,266	91	0.953	55.2	2	2	3	2	3	2	3	3	4	3
3	0	9	21	33	42	50	61	71	82	93	1361.14	1.48	10000	11,363	125	0.952	272.3	6	5	7	9	8	8	9	9	10	11
4	1	9	16	22	29	35	42	49	56	62	1137.3	0.21	6880	8,018	86	0.950	7.7	0	1	1	0	0	1	1	1	1	0
5	0	5	11	17	22	25	31	36	42	48	717.18	0.89	5040	5,758	63	0.954	53.1	3	3	4	5	5	4	5	5	6	7

Table A1-10: Multi-Period Results, Trough Demand Pattern

Problem	Period	Part Order Quantities										Costs				Q^d	Svc. Level	Time (s)	Excess by Period (units)									
		1	2	3	4	5	6	7	8	9	10	Purchase	Holding	Disassembly	Total				1	2	3	4	5	6	7	8	9	10
1	1	0	2	5	8	12	17	21	24	27	31	1712.92	1.894176	8400	10,115	105	0.951	204.1	9	8	7	7	7	8	8	8	7	8
1	2	0	1	4	6	10	13	16	19	21	25	1334.46	0.187948	6640	7,975	83	0.971	5.7	1	0	0	0	0	0	0	0	0	1
1	3	0	3	4	6	9	11	14	16	18	19	1189.38	1.111271	5120	6,310	64	0.950	40.5	4	5	4	4	5	4	4	5	5	4
1	4	0	0	4	7	9	13	17	19	21	25	1330	0.625963	6960	8,291	87	0.950	40.1	4	1	2	3	2	2	2	2	1	3
1	5	0	3	6	8	13	17	21	24	29	30	1774.38	1.337115	8080	9,856	101	0.950	130.3	5	5	5	4	5	5	5	5	7	5
2	1	0	4	8	12	16	21	26	29	32	35	505.05	1.558906	8080	8,587	101	0.951	240.3	5	6	7	8	8	9	10	10	10	10
2	2	0	2	5	8	10	14	17	19	22	24	301.58	0.176829	6560	6,862	82	0.950	5.8	0	0	0	1	0	0	0	0	0	0
2	3	1	3	6	8	11	14	17	19	21	23	413.61	0.9468	4960	5,375	62	0.952	50.1	3	3	4	5	5	6	6	6	6	7
2	4	0	3	5	7	11	15	18	21	24	25	348.95	0.603694	6800	7,150	85	0.952	41.3	2	2	2	2	2	2	2	3	3	2
2	5	0	3	8	11	15	21	25	27	30	34	446.68	1.087424	7920	8,368	99	0.953	154.6	4	4	6	6	6	8	8	7	7	8
3	1	0	8	16	24	33	39	47	56	65	73	6016.12	1.864419	8320	14,338	104	0.951	217.4	8	8	7	7	8	7	7	8	8	8
3	2	0	6	13	19	26	31	38	44	51	58	4766.34	0.180612	6560	11,327	82	0.977	6.4	0	0	0	0	0	0	0	0	0	1
3	3	0	5	10	15	21	24	30	35	40	45	3752.95	1.156455	5200	8,954	65	0.951	48.3	5	5	4	4	5	4	5	5	4	4
3	4	1	7	15	22	27	33	39	46	54	61	5105.73	0.52326	6640	11,746	83	0.950	34.8	1	1	2	2	1	2	1	2	3	3
3	5	0	7	15	23	32	38	47	54	61	70	5781.73	1.45582	8240	14,023	103	0.950	148.9	7	6	6	5	6	6	7	5	4	5
4	1	0	9	19	27	35	41	51	59	69	77	1210.62	1.561675	8080	9,292	101	0.951	243.2	5	6	8	8	8	8	10	10	12	12
4	2	0	6	13	20	26	31	38	44	51	57	860.58	0.172571	6560	7,421	82	0.950	5.3	0	0	0	0	0	0	0	0	0	0
4	3	0	6	12	17	22	26	32	37	43	48	778.46	0.992455	5040	5,819	63	0.954	52.0	3	4	5	5	5	5	6	6	7	7
4	4	0	6	13	20	27	32	40	47	54	61	875.3	0.518338	6800	7,676	85	0.950	41.5	2	1	2	2	2	2	3	3	3	3
4	5	0	9	18	26	34	40	48	56	66	74	1181	1.16653	7920	9,102	99	0.953	159.9	4	5	6	6	6	6	6	6	8	8

REFERENCES

Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs NJ: Prentice Hall (1993).

APICS Dictionary. Alexandria VA: APICS—The Educational Society for Resource Management, 2002.

Bojanowski, Richard S. "Improving Factory Performance with Service Requirements Planning (SRP)," *Production and Inventory Management*, 25:31-44 (1984).

Bruggeman, W. "Maintenance Resource Planning: An Integrative Approach," *Engineering Costs and Production Economics*, 9: 147-155 (1985).

Buchel, Alfred. "Stochastic Material Requirements Planning for Optional Parts," *International Journal of Production Research*, 21: 511-527 (1983).

Collier, David A. "The Measurement and Operating Benefits of Component Part Commonality," *Decision Sciences*, 12: 85-96 (1981).

Darlington, John. "Chasing Muda," *Management Accounting*, XX: 26-30 (1999).

Dowlatshahi, Shad. "Developing a theory of reverse logistics," *Interfaces*, 30: 156-165 (2000).

Epstein, Marc J. "Accounting for Product Take-back," *Management Accounting*, 78: 29-33 (1996).

Ettkin, Lawrence P. and David G. Jahnig. "Adapting MRP II for Maintenance Resource Management Can Provide a Strategic Advantage," *Industrial Engineering*, 18: 50-57 (1986).

Fargher, John S. W.. "Three Shops, three Strategies: Using MRP II, JIT, and TOC in Remanufacturing Cells," *National Productivity Review*, 16: 77-93 (Autumn 1997).

Feeney, G. J. and C. C. Sherbrooke. "The (s-1, s) Inventory Policy Under Compound Poisson Demand," *Management Science*, 12: 391-411 (1966).

Ferrer, Geraldo. "On the Widget Remanufacturing Operation," *European Journal of Operational Research*, 135: 373-XXX (2001).

Ferrer, Geraldo and D. Clay Whybark. "From garbage to goods: Successful remanufacturing systems and skills," *Business Horizons*, 43: 55-64 (2000).

Ferrer, Geraldo and D. Clay Whybark. "Material Planning for a Remanufacturing Facility," *Production and Operations Management*, 10: 112-124 (2001).

Gaudette, Kevin J. *Material Requirements Planning in Air Force Depot-Level Maintenance*. MS Thesis. Graduate School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH (1998).

Ginsburg, Janet. "Once is not Enough," *Business Week*, Issue 3728: 128B-128D (2001).

Giuntini, Ron and Kevin Gaudette. "Remanufacturing: The Next Great Opportunity for Improving U.S. Productivity," *Business Horizons* (to appear, Nov-Dec 2003).

Glover, Fred, Darwin Klingman, and Nancy V. Phillips. *Network Models in Optimization and Their Applications in Practice.* New York: John Wiley & Sons (1992).

Golany, Boaz, Jian Yang, and Gang Yu. "Economic lot-sizing with remanufacturing options," *IIE Transactions,* 33: 995-1003 (2001).

Guide, V. Daniel R., Jr. "Production planning and control for remanufacturing: Industry practice and research needs," *Journal of Operations Management,* 18: 467-483 (2000).

Guide, V. Daniel R., Jr., Mark E. Kraus, and Rajesh Srivastava. "Scheduling policies for remanufacturing," *International Journal of Production Economics,* 48: 187-204 (1997).

Guide, V. Daniel R., Jr., and Rajesh Srivastava. "Buffering from material recovery uncertainty in a recoverable manufacturing environment," *Journal of the Operational Research Society,* 48: 519-529 (1997[a]).

Guide, V. Daniel R., Jr., and Rajesh Srivastava. "An evaluation of order release strategies in a remanufacturing environment," *Computers & Operations Research,* 24: 37-47 (1997[b]).

Guide, V. Daniel R., Jr., and Rajesh Srivastava. "Inventory buffers in recoverable manufacturing," *Journal of Operations Management,* 16: 551-569 (1998).

Guide, V. Daniel R., Jr., Rajesh Srivastava, and Mark E. Kraus. "Proactive expediting policies for recoverable manufacturing," *Journal of the Operational Research Society,* 49: 479-491 (1998).

Guide, V. Daniel R., Jr., Rajesh Srivastava, and Michael S. Spencer. "Are Production Systems Ready for the Green Revolution?" *Production and Inventory Management Journal,* 70-76 (Fourth Quarter 1996).

Guide, V. Daniel R., Jr. and Luk N. Van Wassenhove. "Managing Product Returns for Remanufacturing," *Production and Operations Management,* 10: 142-155 (2001).

Gupta, S.M. and K.N. Taleb. "Scheduling disassembly," *International Journal of Production Research,* 32: 1857-1866 (1994).

Gurnani, Haresh, Ram Akella, and John Lehoczky. "Supply Management in assembly systems with random yield and random demand," *IIE Transactions,* 32: 701-714 (2000).

Hegseth, M.A. "The Challenge of Operation Yield," *Production and Inventory Management Journal,* 25: 4-10 (1984).

Hormozi, Amir M. "Parts remanufacturing in the automotive industry," *Production and Inventory Management Journal,* 38: 26-31 (1997).

Hormozi, Amir M. "Make it Again," *IIE Solutions,* XX: 39-42 (1999).

Inderfurth, Karl, A. G. de Kok, and S. D. P. Flapper. "Product recovery in stochastic remanufacturing systems with multiple reuse options," *European Journal of Operational Research*, 133: 130- (2001).

Inderfurth, Karl and Erwin van der Laan. "Leadtime effects and policy improvement for stochastic inventory control with remanufacturing," *International Journal of Production Economics*, 71: 381-390 (2001).

Jayaraman, V., VDR Guide Jr., and R. Srivastava. "A closed-loop logistics model for remanufacturing," *Journal of the Operational Research Society*, 50: 497-508 (1999).

Kelle, Peter and Edward A. Silver. "Purchasing Policy of New Containers Considering the Random Returns of Previously Issued Containers," *IIE Transactions*, 21: 349-354 (1989).

Kiesmuller, Gudrun P. and Erwin A. van der Laan. "An inventory model with dependent product demands and returns," *International Journal of Production Economics*, 72: 73-87 (2001).

Klausner, Markus and Chris T. Hendrickson. "Reverse-logistics strategy for product take-back," *Interfaces*, 30: 156-165 (2000).

Krupp, James A. G. "Core Obsolescence Forecasting in Remanufacturing," *Production and Inventory Management Journal*, 33: 12-17 (1992).

Krupp, James A. G. "Structuring Bills of Materials for Automotive Remanufacturing," *Production and Inventory Management Journal*, 34: 46-52 (1993).

Kurtulis, I. And D.W. Pentico. "Material Requirements Planning When There Is Scrap Loss," *Production and Inventory Management*, 29: 18-21 (1988).

Lasserre, J. B., C. Bes, and F. Roubellat. "The Stochastic Discrete Dynamic Lot Siz Problem: An Open-Loop Solution," *Operations Research*, 33: 684-689 (1985).

Linton, Jonathan D. and David A. Johnston. "A Decision Support System for Planning Remanufacturing at Nortel Networks," *Interfaces*, 30: 17-31 (2000).

Lund, Robert T. "Remanufacturing," *Technology Review*, 19-29 (February/March 1984).

Lund, Robert T. "The Remanufacturing Industry: Hidden Giant," Boston: Boston University (1996).

Mabert, Vincent A. "A Case Study of Encoder Shift Scheduling Under Uncertainty," *Management Science*, 25: 623-630 (1979).

Majumder, Pranab and Harry Groenevelt. "Competition in Remanufacturing," *Production and Operations Management*, 10: 125-141 (2001).

McClenahan, John S. "Remanufacturing America," *Industry Week*, 251: 34-41 (2002).

Meal, H.C. "Safety Stocks in MRP Systems," Cambridge MA: Massachusetts Institute of Technology, Operations Research Center. Technical Report #166 (1979).

Merrle, Thomas, Jr. "Emerging Technology: Production Scheduling Matures," *IIE Solutions*, 29: 24-27 (1997).

Muckstadt, J. A. and M. H. Isaac, "An analysis of single item inventory system returns," *Naval Research Logistics Quarterly*, 28: 237-254 (1981).

New, C. and J. Mapes. "MRP with High Uncertain Yield Losses," *Journal of Operations Management*, 4: 315-330 (1984).

Newman, Richard G. "MRP Where M = Preventative Maintenance," *Production and Inventory Management*, 26: 21-29 (1985).

O'Brien, Christopher. "Sustainable production – a new paradigm for a new millennium," *International Journal of Production Economics*, 60-61: 1-7 (1999).

Orlicky, J. *Material Requirements Planning*. New York: McGraw Hill (1975).

Panisset, Brian D. "MRP II for Repair/Refurbish Industries," *Production and Inventory Management Journal*, 29: 12-15 (1988).

Parlar, Mahmut. "A Stochastic Production Planning Model with a Dynamic Chance Constraint," *European Journal of Operational Research*, 20: 255-260 (1985).

Perry, James H. "The Impact of Lot Size and Production Scheduling on Inventory Investment in a Remanufacturing Environment," *Production and Inventory Management Journal*, 32: 41-45 (1991).

Rardin, Ronald L. *Optimization in Operations Research*, New Jersey: Prentice Hall (1998).

Richter, Knut and Imre Dobos. "Analysis of the EOQ repair and waste disposal problem with integer setup numbers," *International Journal of Production Economics*, 59: 463-467 (1999).

Richter, Knut and Mirko Sombrutzki. "Remanufacturing planning for the reverse Wagner/Whitin models," *European Journal of Operational Research*, 121: 304- (2000).

Richter, Knut and Jens Weber. "The reverse Wagner/Whitin model with variable manufacturing and remanufacturing cost," *International Journal of Production Economics*, 71: 447-456 (2001).

Sherbrooke, C. C. "METRIC: A Multi-Echelon Technique for Recoverable Item Control," *Operations Research*, 16: 122-141 (1968).

Sherbrooke, C. C. "An Evaluator for the Number of Operationally Ready Aircraft in a Multi-Level Supply System," *Operations Research*, 19: 618-635 (1971).

Sherbrooke, C. C. *Optimal Inventory Modeling of Systems*, New York: John Wiley & Sons (1992).

Souza, Gilvan C., Michael E. Ketzenberg, and V. Daniel R. Guide, Jr. "Capacitated remanufacturing with service level constraints," *Production and Operations Management*, 11: 231-248 (2002).

Teunter, Ruud and Erwin van der Laan. "On the non-optimality of the average cost approach for inventory models with remanufacturing," *International Journal of Production Economics*, 79: 67-73 (2002).

Teunter, Ruud, Erwin van der Laan, and Karl Inderfurth. "How to set the holding cost rates in average cost inventory models with reverse logistics?" *Omega*, 28: 409-415 (2000).

Teunter, Ruud and Dimitrios Vlachos. "On the necessity of a disposal option for returned items that can be remanufactured," *International Journal of Production Economics*, 75: 257-266 (2002).

Thierry, Martijn, Marc Salomon, Jo Van Nunen, and Luk Van Wassenhove. "Strategic Issues in Product Recovery Management," *California Management Review*, 37: 114-135 (1995).

Toktay, L. Beril, Lawrence M. Wein, and Stefanos A. Zenios. "Inventory Management of Remanufacturable Products," *Management Science*, 46: 1412-1426 (2000).

van der Laan, Erwin, Rommert Dekker, and Marc Salomon. "Product remanufacturing and disposal: A numerical comparison of alternative control strategies," *International Journal of Production Economics*, 45: 489-498 (1996).

van der Laan, Erwin and Marc Salomon. "Production planning and inventory control with remanufacturing and disposal," *European Journal of Operational Research*, 102: 264-278 (1997).

van der Laan, Erwin, Marc Salomon, and Rommert Dekker. "An investigation of lead-time effects in manufacturing/remanufacturing systems under simple PUSH and PULL control strategies," *European Journal of Operational Research*, 115: 195-214 (1999).

van der Laan, Erwin, Marc Salomon, Rommert Dekker, and Ad Ridder. "An (s,Q) inventory model with remanufacturing and disposal," *International Journal of Production Economics*, 46/47: 339-350 (1996).

van der Laan, Erwin, Marc Salomon, Rommert Dekker, and Luk Van Wassenhove. "Inventory Control in Hybrid Systems with Remanufacturing," *Management Science,* 45: 733-747 (1999).

Wacker, John G. and Mark Treleven. "Component Part Standardization: An Analysis of Commonality Sources and Indices," *Journal of Operations Management*, 6: 219-244 (1986).

Wassweiler, William. "Know the MRP II Fundamentals that Support Remanufacturing," *Manufacturing Systems*, 11: 52-54 (1993).

Zinn, Walter and Howard Marmorstein. "Comparing Two Alternative Methods of Determining Safety Stock Levels: The Demand and Forecast Systems," *Journal of Business Logistics*, 11: 95-110 (1990).